

一、环境

1. linux版本: CentOS release 6.8 (Final)
2. mysql版本: mysql-5.7.30-1.el6.x86_64.rpm-bundle.tar
3. 架构
 - 4台主机, 1主、2从、1 MHA
 - master: 192.168.238.156
 - slave: 192.168.238.157, 192.168.238.158
 - MHA: 192.168.238.159

二、mysql安装

1. 移除原系统mysql和mariadb

- 1) 查看系统是否安装mariadb和mysql

```
mariadb-rpm -qa | grep -i mariadb
```

```
rpm -qa | grep -i mysql
```

- 2) 移除

```
rpm -e MySQL-devel-5.6.34-1.el6.i686 --nodeps  
rpm -e MySQL-server-5.6.34-1.el6.i686 --nodeps  
rpm -e MySQL-client-5.6.34-1.el6.i686 --nodeps
```

2. rpm方式安装mysql

- 1) 解压缩mysql tar包

```
tar -xvf mysql-5.7.30-1.el6.x86_64.rpm-bundle.tar
```

- 2) 依次安装common/libs/libs-compat/client/server/devel rpm

```
rpm -ivh mysql-community-common-5.7.30-1.el6.x86_64.rpm  
rpm -ivh mysql-community-libs-5.7.30-1.el6.x86_64.rpm  
rpm -ivh mysql-community-libs-compat-5.7.30-1.el6.x86_64.rpm  
rpm -ivh mysql-community-client-5.7.30-1.el6.x86_64.rpm  
rpm -ivh mysql-community-server-5.7.30-1.el6.x86_64.rpm  
rpm -ivh mysql-community-devel-5.7.30-1.el6.x86_64.rpm
```

- 3) 初始化mysql

```
mysqld --initialize --user=mysql
```

- 4) 查看mysql初始化密码

```
cat /var/log/mysqld.log
```

- 5) 启动mysql

```
service mysql start
```

6)登录重置密码

```
set password=password('root');
```

7)重新登录

三、mysql主从环境搭建

1.主库配置 (master:192.168.238.156)

1)修改 /etc/my.cnf

```
vim /etc/my.cnf
```

```
[mysqld]
log_bin=mysql-bin
server-id=1
sync-binlog=1

#对哪些库不同步
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
binlog-ignore-db=sys

#对哪些库同步
#binlog-do-db=lagou
```

2)重启mysql

```
service mysqld restart
```

3)登录mysql授权

```
grant replication slave on *.* to 'root'@'%' identified by 'root';
grant all privileges on *.* to 'root'@'%' identified by 'root';
flush privileges;
```

4)查看主库状态

```
show master status
```

```
mysql> show master status;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 |      869 |              | information_schema,performance_schema,sys |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2.从库配置 (slave: 192.168.238.157)

1)修改 /etc/my.cnf

```
vim /etc/my.cnf
```

```
[mysqld]
server-id=2
relay_log=mysql-relay-bin
##只读配置
#read_only=1
```

2)重启mysql

service mysqld restart

3)登录mysql查看从库状态

show slave status \G

4)同步主库

change master to

master_host='192.168.238.156',master_port=3306,master_user='root',master_password='root',master_log_file='mysql-bin.000001',master_log_pos=869;

5)开启从库

start slave;

6)查看从库状态

show slave status \G

```
mysql> show slave status \G
Empty set (0.00 sec)

mysql> change master to master_host='192.168.238.156',master_port=3306,master_user='root',master_password='root',master_log_file='mysql-bin.000001',master_log_pos=869;
Query OK, 0 rows affected, 2 warnings (0.09 sec)

mysql> start slave;
Query OK, 0 rows affected (0.05 sec)

mysql> show slave status \G
+-----+
| Slave_IO_State: Checking master version |
| Master_Host: 192.168.238.156 |
| Master_User: root |
| Master_Port: 3306 |
| Connect_Retry: 60 |
| Master_Log_File: mysql-bin.000001 |
| Read_Master_Log_Pos: 869 |
| Relay_Log_File: mysql-relay-bin.000001 |
| Relay_Log_Pos: 4 |
| Relay_Master_Log_File: mysql-bin.000001 |
| 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 |
+-----+
```

3.从库配置 (slave: 192.168.238.158)

同从库(192.168.238.157)配置， /etc/my.cnf 中server-id=3

4.测试主从配置

主库执行sql，查看从库是否同步

```
create database lagou;
show database;
use lagou;
create table position (id int primary key auto_increment, name varchar(20),
salary varchar(20), city varchar(20)) engine=innodb charset=utf8;
create table position_detail (id int primary key auto_increment, pid int,
description text) engine=innodb charset=utf8;
```

```
ssh://root:*****@192.168.238.156:22
要添加当前会话，点击左侧的箭头按钮。

1 centos_192.168.238.... × 2 centos_192.168.238.157 × 3 centos_192.168.238.158 × +

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

mysql> use lagou;
Database changed
mysql> create table position (id int primary key auto_increment, name varchar(20), salary varchar(20), city varchar(20)) engine=innodb charset=utf8;
Query OK, 0 rows affected (0.09 sec)

mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position         |
+-----+
1 row in set (0.00 sec)

mysql> create table position_detail (id int primary key auto_increment, pid int, description text) engine=innodb charset=utf8;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position         |
| position_detail  |
+-----+
2 rows in set (0.01 sec)
```

```
ssh://root:*****@192.168.238.157:22
要添加当前会话，点击左侧的箭头按钮。

1 centos_192.168.238.156 × 2 centos_192.168.238.... × 3 centos_192.168.238.158 × +

Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0.00 sec)

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

mysql> use lagou;
Database changed
mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position         |
+-----+
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position         |
| position_detail  |
+-----+
2 rows in set (0.00 sec)
```

```
ssh://root:*****@192.168.238.158:22
要添加当前会话，点击左侧的箭头按钮。
1 centos_192.168.238.156 × 2 centos_192.168.238.157 × 3 centos_192.168.238.... × +

Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0.00 sec)

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

mysql> use lagou;
Database changed
mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position |
+-----+
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_lagou |
+-----+
| position |
| position_detail |
+-----+
2 rows in set (0.00 sec)
```

5.半同步复制机制配置

主库配置

- 1)登录mysql后查看是否有动态加载
`select @@have_dynamic_loading;`
- 2)查看插件
`show plugins;`
- 3)安装插件
`install plugin rpl_semi_sync_master soname 'semisync_master.so';`
- 4)查看
`show variables like '%semi%';`
- 5)设置rpl_semi_sync_master_enabled和rpl_semi_sync_master_timeout
`set global rpl_semi_sync_master_enabled=1;`
`set global rpl_semi_sync_master_timeout=1000;`

```
mysql> install plugin rpl_semi_sync_master soname 'semisync_master.so';
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> show variables like '%semi%';
```

Variable_name	Value
rpl_semi_sync_master_enabled	OFF
rpl_semi_sync_master_timeout	10000
rpl_semi_sync_master_trace_level	32
rpl_semi_sync_master_wait_for_slave_count	1
rpl_semi_sync_master_wait_no_slave	ON
rpl_semi_sync_master_wait_point	AFTER_SYNC

```
6 rows in set (0.05 sec)
```

```
mysql> set global rpl_semi_sync_master_enabled=1;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> set global rpl_semi_sync_master_timeout=1000;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%semi%';
```

Variable_name	Value
rpl_semi_sync_master_enabled	ON
rpl_semi_sync_master_timeout	1000
rpl_semi_sync_master_trace_level	32
rpl_semi_sync_master_wait_for_slave_count	1
rpl_semi_sync_master_wait_no_slave	ON
rpl_semi_sync_master_wait_point	AFTER_SYNC

```
6 rows in set (0.01 sec)
```

从库配置

1)安装插件: install plugin rpl_semi_sync_slave soname 'semisync_slave.so';

2)设置rpl_semi_sync_slave_enabled

```
set global rpl_semi_sync_slave_enabled=1;
```

3)重启slave

```
stop slave;
```

```
start slave;
```

```
mysql> install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> show variables like '%semi%';
```

Variable_name	Value
rpl_semi_sync_slave_enabled	OFF
rpl_semi_sync_slave_trace_level	32

```
2 rows in set (0.01 sec)
```

```
mysql> set global rpl_semi_sync_slave_enabled=1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%semi%';
```

Variable_name	Value
rpl_semi_sync_slave_enabled	ON
rpl_semi_sync_slave_trace_level	32

```
2 rows in set (0.00 sec)
```

测试半同步复制

1主库执行sql, 查看从库是否同步

```
insert into position (name, salary, city) values ('lm', '10', '北京');
insert into position_detail(pid, description) values (1, 'haha');
```

查看log日志: tail -200 /var/log/mysqld.log

```
2020-05-03T03:54:29.635475Z 5 [Warning] IP address '192.168.228.159' could not be resolved; Name or service not known
2020-05-03T03:54:29.638952Z 5 [Note] Start binlog dump to master_thread_id(5) slave_server(3), pos(mysql-bin.000001, 869)
2020-05-03T04:09:24.972738Z 3 [Note] Semi-sync replication initialized for transactions.
2020-05-03T04:09:24.972885Z 3 [Note] Semi-sync replication enabled on the master.
2020-05-03T04:09:24.973067Z 0 [Note] Starting ask receiver thread
2020-05-03T04:18:42.410957Z 0 [Note] While initializing dump thread for slave with UUID <78ffd2a3-ad18-11ea-bd23-0050563ecf66>, found a zombie dump thread with the same UUID. Master is killing the zombie dump thread(4).
2020-05-03T04:18:42.411135Z 4 [Note] Stop asynchronous binlog dump to slave (server_id: 2)
2020-05-03T04:18:42.411055Z 6 [Note] Start binlog dump to master_thread_id(6) slave_server(2), pos(mysql-bin.000001, 1571)
2020-05-03T04:18:42.411782Z 6 [Note] Start semi-sync binlog dump to slave (server_id: 2), pos(mysql-bin.000001, 1571)
2020-05-03T04:18:50.330973Z 7 [Note] While initializing dump thread for slave with UUID <54aae2ab-ad19-11ea-ad7d-005056294c78>, found a zombie dump thread with the same UUID. Master is killing the zombie dump thread(5).
2020-05-03T04:18:50.331158Z 5 [Note] Stop asynchronous binlog dump to slave (server_id: 3)
2020-05-03T04:18:50.331685Z 7 [Note] Start binlog dump to master_thread_id(7) slave_server(3), pos(mysql-bin.000001, 1571)
2020-05-03T04:18:50.331703Z 7 [Note] Start semi-sync binlog dump to slave (server_id: 3), pos(mysql-bin.000001, 1571)
root@localhost:~# tail -200 /var/log/mysqld.log
```