目录

[导言 2](#_Toc490656518)

[MySQL安装 2](#_Toc490656519)

[主主复制搭建 6](#_Toc490656520)

[keepalived 配置 9](#_Toc490656521)

[问题&讨论 12](#_Toc490656522)

## 导言

实验环境说明：

|  |  |
| --- | --- |
| OS版本 | CentOS7 |
| MySQL版本 | mysql-5.7.14 |
| Keepalived | ISO 自带 keepalived-1.2.13-7.el7.x86\_64 |

服务器说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名称 | IP | 配置 | 用途 |
| c1 | 192.168.137.201 | 1核1G | 主节点之一，需安装MySQL、  Keepalived |
| c2 | 192.168.137.202 | 1核1G | 主节点之一，需安装MySQL、  Keepalived |

## MySQL安装

* **安装方法：二进制安装**
* **安装包：mysql-5.7.14-linux-glibc2.5-x86\_64.tar.gz**
* **c1服务器my.cnf**

[client]

port    = 3306

socket    = /tmp/mysql.sock

[mysql]

prompt="\u@c1 \R:\m:\s [\d]> "

no-auto-rehash

[mysqld]

user    = mysql

port    = 3306

basedir    = /usr/local/mysql

datadir    = /data/mysql/mysql3306/data

socket    = /tmp/mysql.sock

pid-file = c1.pid

character-set-server = utf8mb4

skip\_name\_resolve = 1

open\_files\_limit    = 65535

back\_log = 1024

max\_connections = 20

max\_connect\_errors = 1000000

table\_open\_cache = 100

table\_definition\_cache = 100

table\_open\_cache\_instances = 64

thread\_stack = 512K

external-locking = FALSE

max\_allowed\_packet = 32M

sort\_buffer\_size = 4M

join\_buffer\_size = 4M

thread\_cache\_size = 30

query\_cache\_size = 0

query\_cache\_type = 0

interactive\_timeout = 600

wait\_timeout = 600

tmp\_table\_size = 32M

max\_heap\_table\_size = 32M

slow\_query\_log = 1

slow\_query\_log\_file = /data/mysql/mysql3306/slow.log

log-error = /data/mysql/mysql3306/error.log

long\_query\_time = 1

server-id = 103306

log-bin = /data/mysql/mysql3306/logs/mysql-bin

sync\_binlog = 1

binlog\_cache\_size = 4M

max\_binlog\_cache\_size = 2G

max\_binlog\_size = 1G

expire\_logs\_days = 7

master\_info\_repository = TABLE

relay\_log\_info\_repository = TABLE

gtid\_mode = on

enforce\_gtid\_consistency = 1

log\_slave\_updates

binlog\_format = row

relay\_log\_recovery = 1

relay-log-purge = 1

key\_buffer\_size = 32M

read\_buffer\_size = 8M

read\_rnd\_buffer\_size = 4M

bulk\_insert\_buffer\_size = 64M

myisam\_sort\_buffer\_size = 128M

myisam\_max\_sort\_file\_size = 10G

myisam\_repair\_threads = 1

lock\_wait\_timeout = 3600

explicit\_defaults\_for\_timestamp = 1

innodb\_thread\_concurrency = 0

innodb\_sync\_spin\_loops = 100

innodb\_spin\_wait\_delay = 30

transaction\_isolation = REPEATABLE-READ

#innodb\_additional\_mem\_pool\_size = 16M

innodb\_buffer\_pool\_size = 717M

innodb\_buffer\_pool\_instances = 8

innodb\_buffer\_pool\_load\_at\_startup = 1

innodb\_buffer\_pool\_dump\_at\_shutdown = 1

innodb\_data\_file\_path = ibdata1:1G:autoextend

innodb\_flush\_log\_at\_trx\_commit = 1

innodb\_log\_buffer\_size = 32M

innodb\_log\_file\_size = 2G

innodb\_log\_files\_in\_group = 2

innodb\_max\_undo\_log\_size = 4G

innodb\_io\_capacity = 4000

innodb\_io\_capacity\_max = 8000

innodb\_flush\_neighbors = 0

innodb\_write\_io\_threads = 8

innodb\_read\_io\_threads = 8

innodb\_purge\_threads = 4

innodb\_page\_cleaners = 4

innodb\_open\_files = 65535

innodb\_max\_dirty\_pages\_pct = 50

innodb\_flush\_method = O\_DIRECT

innodb\_lru\_scan\_depth = 4000

innodb\_checksum\_algorithm = crc32

#innodb\_file\_format = Barracuda

#innodb\_file\_format\_max = Barracuda

innodb\_lock\_wait\_timeout = 10

innodb\_rollback\_on\_timeout = 1

innodb\_print\_all\_deadlocks = 1

innodb\_file\_per\_table = 1

innodb\_online\_alter\_log\_max\_size = 4G

internal\_tmp\_disk\_storage\_engine = InnoDB

innodb\_stats\_on\_metadata = 0

innodb\_status\_file = 1

innodb\_status\_output = 0

innodb\_status\_output\_locks = 0

#performance\_schema

performance\_schema = 1

performance\_schema\_instrument = '%=on'

#innodb monitor

innodb\_monitor\_enable="module\_innodb"

innodb\_monitor\_enable="module\_server"

innodb\_monitor\_enable="module\_dml"

innodb\_monitor\_enable="module\_ddl"

innodb\_monitor\_enable="module\_trx"

innodb\_monitor\_enable="module\_os"

innodb\_monitor\_enable="module\_purge"

innodb\_monitor\_enable="module\_log"

innodb\_monitor\_enable="module\_lock"

innodb\_monitor\_enable="module\_buffer"

innodb\_monitor\_enable="module\_index"

innodb\_monitor\_enable="module\_ibuf\_system"

innodb\_monitor\_enable="module\_buffer\_page"

innodb\_monitor\_enable="module\_adaptive\_hash"

[mysqldump]

quick

max\_allowed\_packet = 32M

* **c2服务器my.cnf**

**与c1有差异的两个参数**

pid-file = c2.pid

server-id = 203306

* **安装结果**

具体mysql二进制安装方法不再介绍

c1实例安装完毕，并且开启了GTID

[root@c1 ~]# mysql -uroot -proot

mysql> show global variables like '%gtid%';

+----------------------------------+------------------------------------------+

| Variable\_name | Value |

+----------------------------------+------------------------------------------+

| binlog\_gtid\_simple\_recovery | ON |

| enforce\_gtid\_consistency | ON |

| gtid\_executed | 0cc853e5-656d-11e7-b57d-000c29ef5e6c:1-4 |

| gtid\_executed\_compression\_period | 1000 |

| gtid\_mode | ON |

| gtid\_owned | |

| gtid\_purged | |

| session\_track\_gtids | OFF |

+----------------------------------+------------------------------------------+

8 rows in set (0.00 sec)

c2实例也安装好了

[root@c2 data]# mysql -uroot -proot

mysql> show global variables like '%gtid%';

+----------------------------------+------------------------------------------+

| Variable\_name | Value |

+----------------------------------+------------------------------------------+

| binlog\_gtid\_simple\_recovery | ON |

| enforce\_gtid\_consistency | ON |

| gtid\_executed | 49c226b7-6589-11e7-9037-000c29bfa54a:1-4 |

| gtid\_executed\_compression\_period | 1000 |

| gtid\_mode | ON |

| gtid\_owned | |

| gtid\_purged | |

| session\_track\_gtids | OFF |

+----------------------------------+------------------------------------------+

8 rows in set (0.00 sec)

## 主主复制搭建

* **c1实例创建复制用户**

mysql> create user 'repl'@'192.168.137.202' identified by 'repl';

Query OK, 0 rows affected (0.00 sec)

mysql> grant replication slave on \*.\* to 'repl'@'192.168.137.202';

Query OK, 0 rows affected (0.05 sec)

* **c2实例创建复制用户**

mysql> create user 'repl'@'192.168.137.201' identified by 'repl';

Query OK, 0 rows affected (0.00 sec)

mysql> grant replication slave on \*.\* to 'repl'@'192.168.137.201';

Query OK, 0 rows affected (0.00 sec)

* **c2实例同步c1实例**

c2 上

mysql> CHANGE MASTER TO

    ->   MASTER\_HOST='192.168.137.201',

    ->   MASTER\_USER='repl',

    ->   MASTER\_PASSWORD='repl',

    ->   MASTER\_PORT=3306,

    ->   master\_auto\_position=1;

Query OK, 0 rows affected, 2 warnings (0.08 sec)

mysql> start slave;

Query OK, 0 rows affected (0.00 sec)

mysql> show slave status\G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

               Slave\_IO\_State: Waiting for master to send event

                  Master\_Host: 192.168.137.201

                  Master\_User: repl

                  Master\_Port: 3306

                Connect\_Retry: 60

              Master\_Log\_File: mysql-bin.000005

          Read\_Master\_Log\_Pos: 1260

               Relay\_Log\_File: c2-relay-bin.000002

                Relay\_Log\_Pos: 1433

        Relay\_Master\_Log\_File: mysql-bin.000005

             Slave\_IO\_Running: Yes

            Slave\_SQL\_Running: Yes

......

             Master\_Server\_Id: 103306

                  Master\_UUID: 0cc853e5-656d-11e7-b57d-000c29ef5e6c

             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 more updates

           Master\_Retry\_Count: 86400

                  Master\_Bind:

      Last\_IO\_Error\_Timestamp:

     Last\_SQL\_Error\_Timestamp:

               Master\_SSL\_Crl:

           Master\_SSL\_Crlpath:

           Retrieved\_Gtid\_Set: 0cc853e5-656d-11e7-b57d-000c29ef5e6c:5-9

            Executed\_Gtid\_Set: 0cc853e5-656d-11e7-b57d-000c29ef5e6c:1-9,

49c226b7-6589-11e7-9037-000c29bfa54a:1-2

                Auto\_Position: 1

* **c1实例同步c2实例**

mysql> CHANGE MASTER TO

    ->   MASTER\_HOST='192.168.137.202',

    ->   MASTER\_USER='repl',

    ->   MASTER\_PASSWORD='repl',

    ->   MASTER\_PORT=3306,

    ->   master\_auto\_position=1;

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: 192.168.137.202

                  Master\_User: repl

                  Master\_Port: 3306

                Connect\_Retry: 60

              Master\_Log\_File: mysql-bin.000010

          Read\_Master\_Log\_Pos: 1729

               Relay\_Log\_File: c1-relay-bin.000002

                Relay\_Log\_Pos: 883

        Relay\_Master\_Log\_File: mysql-bin.000010

             Slave\_IO\_Running: Yes

            Slave\_SQL\_Running: Yes

...

             Master\_Server\_Id: 203306

                  Master\_UUID: 49c226b7-6589-11e7-9037-000c29bfa54a

             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 more updates

           Master\_Retry\_Count: 86400

                  Master\_Bind:

      Last\_IO\_Error\_Timestamp:

     Last\_SQL\_Error\_Timestamp:

               Master\_SSL\_Crl:

           Master\_SSL\_Crlpath:

           Retrieved\_Gtid\_Set: 49c226b7-6589-11e7-9037-000c29bfa54a:1-2

            Executed\_Gtid\_Set: 0cc853e5-656d-11e7-b57d-000c29ef5e6c:1-9,

49c226b7-6589-11e7-9037-000c29bfa54a:1-2

                Auto\_Position: 1

此时双主搭建完毕！

* **注意点**

## keepalived 配置

* **c1服务器**

**安装keepalived**

yum -y install keepalived

yum install net-tools

**配置keepalived.conf**

vi keepalived.conf

! Configuration File for keepalived

global\_defs {

   router\_id LVS\_mysql201

}

vrrp\_script vs\_mysql\_201 {

    script "/etc/keepalived/check.sh"

    interval 1

    weight -10

}

vrrp\_instance VI\_1 {

    state MASTER

    interface ens33

    virtual\_router\_id 60

    priority 100

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    unicast\_src\_ip  192.168.137.201

    unicast\_peer {

                   192.168.137.202

                 }

    virtual\_ipaddress {

        192.168.137.77/24       brd 192.168.137.255       dev ens33 label ens33:vip

     }

track\_script {

        vs\_mysql\_201

    }

}

**配置检查脚本check.sh**

#!/bin/bash

#

daemonid=`ps -ef |grep "/usr/local/mysql/bin/mysqld --basedir" |grep -v grep|awk '{print $2}'`

if [ "$daemonid" != "" ];then

   print 0

   exit 0

else

   service keepalived stop

fi

* **c2服务器**

**安装keepalived**

yum -y install keepalived

yum install net-tools

**配置keepalived.conf**

vi keepalived.conf

! Configuration File for keepalived

global\_defs {

   router\_id LVS\_mysql202

}

vrrp\_script vs\_mysql\_202 {

    script "/etc/keepalived/check.sh"

    interval 1

    weight -10

}

vrrp\_instance VI\_1 {

    state MASTER

    interface ens33

    virtual\_router\_id 60

    priority 95

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    unicast\_src\_ip  192.168.137.202

    unicast\_peer {

                   192.168.137.201

                 }

    virtual\_ipaddress {

        192.168.137.77/24       brd 192.168.137.255       dev ens33 label ens33:vip

     }

track\_script {

        vs\_mysql\_202

    }

}

**配置检查脚本check.sh**

#!/bin/bash

#

daemonid=`ps -ef |grep "/usr/local/mysql/bin/mysqld --basedir" |grep -v grep|awk '{print $2}'`

if [ "$daemonid" != "" ];then

   print 0

   exit 0

else

   service keepalived stop

fi

* **keepalived的脑裂问题**

**解决思路如下：**

**每个keepalived的节点都执行一个定时任务的脚本，定时去ping网关，累计失败次数超过阀值次数，则关闭自身的keepalived服务。这样就不会出现脑裂的情况。**

**脚本如下：**

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

import os

import sys

ip = sys.argv[1]

c=0

for i in range(1,4,1):

return1=os.popen('ping %s -w 1' %ip).read()

if '100% packet loss' in return1 :

c += 1

if c>2:

os.popen('service keepalived stop')

**定时任务如下：**

Shell>crontab –e

\* \* \* \* \* /root/ping.py 192.168.137.1

## 问题&讨论

* **问题一**

如果c2实例是通过c1实例虚拟机复制而来，server UUID就是一样的，这时候搭建主从就会报错，可以删除datadir下的auto.cnf，重启实例就会生成新的UUID

* **问题二**

Keepalived的MySQL进程监控脚本中，对进程的准确获取很重要，需要根据自身具体的启动方式来确定

* **问题三**

Keepalived的脑裂处理脚本中，对Python执行ping命令后的结果的获取和判断，需要考虑到现有环境的字符集问题，特别是使用Python2的