

mysql 双主 keepalived GTID

参考：<http://www.cnblogs.com/liangshaoye/p/5459421.html>

<http://www.cnblogs.com/icyblog/p/7241040.html>

<http://keithlan.github.io/2016/06/23/gtid/>

一、搭建mysql双主GTID复制

1. 修改主服务器mysql配置文件/etc/my.cnf，添加以下：（注意不要重复定义参数）

```
server-id      = 201          //这里的ID号与从库上或者主库上的ID必须保证不一样
log-slave-updates=true       //slave 更新是否记入日志
gtid-mode=on                // 启用gtid类型，否则就是普通的复制架构
enforce-gtid-consistency=true //强制GTID 的一致性
master-info-repository=TABLE //主服信息记录库=表 /文件
relay-log-info-repository=TABLE //中继日志信息记录库
sync-master-info=1          //同步主库信息
slave-parallel-workers=4     //从服务器的SQL 线程数，要复制库数目相同
binlog-checksum=CRC32        // 校验码，可以自定义
master-verify-checksum=1     //主服校验
slave-sql-verify-checksum=1  //从服校验
binlog-rows-query-log_events=1 //二进制日志详细记录事件
report-port=3310             //提供复制报告端口，当前实例端口号
report-host=192.168.1.31     //提供复制报告主机，本机的ip地址
```

eg：

```
server-id=1
log-slave-updates=true
gtid-mode=on
enforce-gtid-consistency=true
master-info-repository=TABLE
relay-log-info-repository=TABLE
sync-master-info=1
slave-parallel-workers=4
binlog-checksum=CRC32
master-verify-checksum=1
slave-sql-verify-checksum=1
binlog-rows-query-log_events=1
report-port=3306
report-host=10.1.5.233
```

2. 修改从服务器mysql配置文件/etc/my.cnf，增添以下：（注意不要重复定义参数）

```
server-id      = 202          //这里的ID号与从库上或者主库上的ID必须保证不唯一
log-slave-updates=true       //slave 更新是否记入日志
gtid-mode=on                // 启用gtid类型，否则就是普通的复制架构
enforce-gtid-consistency=true //强制GTID 的一致性
master-info-repository=TABLE //主服信息记录库=表 /文件
relay-log-info-repository=TABLE //中继日志信息记录库
sync-master-info=1          //同步主库信息
slave-parallel-workers=4     //从服务器的SQL 线程数，要复制库数目相同
binlog-checksum=CRC32        // 校验码，可以自定义
master-verify-checksum=1     //主服校验
slave-sql-verify-checksum=1  //从服校验
binlog-rows-query-log_events=1 //二进制日志详细记录事件
report-port=3310             //提供复制报告端口，当前实例端口号
report-host=192.168.1.32     //提供复制报告主机，本机的ip地址
```

eg：

```
server-id=2
```

```
log-slave-updates=true
gtid-mode=on
enforce-gtid-consistency=true
master-info-repository=TABLE
relay-log-info-repository=TABLE
sync-master-info=1
slave-parallel-workers=4
binlog-checksum=CRC32
master-verify-checksum=1
slave-sql-verify-checksum=1
binlog-rows-query-log_events=1
report-port=3306
report-host=10.1.5.234
```

2.2 在主库全库导出，并从库应用主库的全备

```
mysqldump --single-transaction --master-data=2 --triggers --routines --all-databases --events -u root -p > backup.sql
```

3. 授权：

A服务器mysql实例3310:

```
grant replication slave,replication client on *.* to slave@'192.168.1.32' identified by '123456';
```

eg：

```
grant replication slave,replication client on *.* to slave@'10.1.5.234' identified by 'Root@911';
```

B服务器mysql实例3310:

```
grant replication slave,replication client on *.* to slave@'192.168.1.31' identified by '123456';
```

eg：

```
grant replication slave,replication client on *.* to slave@'10.1.5.234' identified by 'Root@911';
```

A服务器mysql实例3310:

```
change master to master_host='192.168.1.32',master_port=3310,master_user='slave',master_password='123456',master_auto_position=1;
```

注意：这里要是不配置参数master_port=3311，默认的会去找3306

eg：

```
change master to master_host='10.1.5.234', master_port=3306,master_user=' slave', master_password=' Root@911',
master_auto_position=1;
```

B服务器mysql实例3310:

```
change master to master_host='192.168.1.31',master_port=3310,master_user='slave',master_password='123456',master_auto_position=1;
```

eg：

```
change master to master_host='10.1.5.233', master_port=3306,master_user=' slave', master_password=' Root@911',
master_auto_position=1;
```

A、B服务器上开启主从复制：

```
start slave;
```

4. 查看状态

A服务器： show slave status\G;show processlist;

B服务器： show slave status\G; show processlist;

5、到这里A、B服务器间基于GTID主从复制模式已经配置好了，测试下

在A服务器3310实例上： drop database test01； B服务器3310上查看状态：

```
show databases;
```

在B服务器3310实例上: create database wl_test02; A服务器上3310上查看状态:
show databases;

6. 重置master, slave

```
stop slave;  
reset master;  
reset slave;
```

二、安装配置keepalived负载均衡器

1. 在主, 从服务器上安装keepalived :

wget <http://www.keepalived.org/software/keepalived-1.3.6.tar.gz>

2. 解压缩 :

```
tar -zxvf keepalived-1.3.6.tar.gz
```

3. configure :

```
cd keepalived-1.3.6
```

```
./configure
```

报错 : configure: error: no acceptable C compiler found in \$PATH

解决方法 : `yum groupinstall "Development tools"`

参考 : <https://stackoverflow.com/questions/19816275/no-acceptable-c-compiler-found-in-path-when-installing-python>

继续 configure :

```
./configure
```

报错 :

```
!!! OpenSSL is not properly installed on your system. !!!
```

```
!!! Can not include OpenSSL headers files. !!!
```

解决方法 : `yum install -y openssl-devel.x86_64`

警告 :

*** WARNING - this build will not support IPVS with IPv6. Please install libnl/libnl-3 dev libraries to support IPv6 with IPVS.

解决方法 :

```
yum -y install libnl libnl-devel
```

继续 configure

```
./configure
```

报错 :

configure: error: libnfnetlink headers missing

解决方法 :

```
yum install -y libnfnetlink-devel
```

继续 configure

```
./configure
```

成功!

4. make and make install

```
make&&make install
```

5. `ln -s /usr/local/sbin/keepalived /sbin/`

6. 设置keepalived启动级别

```
chkconfig --level 35 keepalived on
```

7. 配置keepalived

```
vi /usr/local/etc/keepalived/keepalived.conf
```

参考：http://blog.csdn.net/stubborn_cow/article/details/48135771

keepalived配置文件

keepalived.conf配置分为3部分，global_defs, vrrp_instance VI_1, virtual_server。

注：下面蓝色文字是可有可无，红色文字就是需要修改的部分，如果只是测试用，写入virtual_ipaddress, virtual_server, state, priority这四个就可以了。而当主修改好文件后，在从只用修改state, priority, priority不能高于在主设置的值。

```
global_defs {
    notification_email {
        acassen@firewall.loc
        failover@firewall.loc
        sysadmin@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 192.168.200.1
    smtp_connect_timeout 30
    router_id LVS_DEVEL
}
```

- notification_email 故障发生时给谁发邮件通知。
- notification_email_from 通知邮件从哪个地址发出。
- smtp_server 通知邮件的smtp地址。
- smtp_connect_timeout 连接smtp服务器的超时时间。
- router_id 标识本节点的字条串，通常为hostname，但不一定非得是hostname。故障发生时，邮件通知会用到。

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    advert_int 1
    nopreempt
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.200.100
    }
}
```

- state 可以是MASTER或BACKUP，不过当其他节点keepalived启动时会将priority比较大的节点选举为MASTER，因此该项其实没有实质用途。
- interface 节点固有IP（非VIP）的网卡，用来发VRRP包。
- virtual_router_id 取值在0-255之间，用来区分多个instance的VRRP组播。一个网段内，不能存在不同的router id。
- priority 用来选举master的，要成为master，这个选项的值要高于其他机器的值，该项取值范围是1-255（在此范围之外会被识别成默认值100）。
- advert_int 监测间隔(秒)
- nopreempt 防止抢占资源
- authentication 认证区域，认证类型有PASS和HA（IPSEC），推荐使用PASS（密码只识别前8位）。
- virtual_ipaddress vip 虚拟IP，可以多个
- authentication 认证区域，认证类型有PASS和HA（IPSEC），推荐使用PASS（密码只识别前8位）。
- virtual_ipaddress vip 虚拟IP，可以多个

```
virtual_server 192.168.200.100 3306 {
    delay_loop 6
    lb_algo rr
    lb_kind NAT
    nat_mask 255.255.255.0
    persistence_timeout 50
    protocol TCP
    real_server 192.168.201.10 3306 {
        weight 1
        notify_down /usr/local/MySQL/bin/MySQL.sh
    }
    TCP_CHECK {
        connect_port 3306
        connect_timeout 3
        nb_get_retry 3
    }
}
```

```
    delay_before_retry 3
}
}
}
```

- delay_loop 每隔N秒查询realserver状态。
- lb_algo 后端调试算法（load balancing algorithm）。
- lb_kind LVS调度类型NAT/DR/TUN。
- nat_mask 网关。
- persistence_timeout 会话保持时间。
- real_server 真正提供服务的服务器IP。
- weight 权重。
- notify_down 检测到服务down后执行的脚本
- connect_port 连接端口。
- connect_timeout 超时时长。
- nb_get_retry 重试次数
- delay_before_retry 下次重试的时间延迟。

注：notify_down的作用是在mysql停止后，关闭本机的keepalived进程，避免冲突。