Day 03 MySQL多实例 MHA高可用集群

一、MySQL多实例

1.多实例概述

·什么是多实例

- 在一台物理主机上运行多个数据库服务

·问什么要使用多实例

- 节约运维成本

- 提高硬件利用率

5.7.17不支持多实例

# ls mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz

2.配置多实例

1）配置步骤

- 安装支持多实例服务的软件包

·解压

·修改目录名

·修改PATH变量

# tar -xf mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz

# mv mysql-5.7.20-linux-glibc2.12-x86\_64 /usr/local/mysql

# tail -1 /etc/profile

export PATH=/usr/local/mysql/bin:$PATH

# source /etc/profile

- 修改主配置文件

- 根据配置文件做相应设置

- 初始化授权库

- 启动服务

- 客户端访问

2）配置文件参数说明

·主配置文件/etc/my.cnf

- 每个实例要有独立的数据库目录和监听端口号

- 每个实例要有独立的实例名称和独立的sock文件

[mysqld\_multi] //启用多实例

mysqld = /usr/local/mysql/bin/mysqld\_safe //指定进程文件的路径

mysqladmin = /usr/local/mysql/bin/mysqladmin //指定管理命令路径

user = root //指定调用进程的用户

[mysqldX] //实例进程名称，X标识实例名称如[mysql2]

port = 3307 //端口号

datadir = /data3307 //数据库目录，要手动创建

socket = /data3307/mysql.sock //指定sock文件的路径和名称

pid-file = /data3307/mysqld.pid //进程pid号文件位置

log-error = /data3307/mysqld.err //错误日志位置

3）管理多实例

# useradd mysql

# groupadd mysql

# mkdir /data3307 /data3308

•会提示root用户登录的初始化密码

•启动实例进程

•停止实例进程

# mysqld --user=mysql --basedir=/usr/local/mysql --datadir=/data3307 --initialize

•查看是否启动

# netstat -antulp | grep 3307

# ss -antulp | grep 3308

•启动服务

# mysqld\_multi start 1

# mysqld\_multi start 2

•验证服务启动情况

# netstat -antulp | grep 3307

# ss -antulp | grep 3308

# ps aux | grep mysqld

# ps -C mysqld

•停止进程

# mysqld\_multi --user=root --passwor=密码 stop 实例编号

4）客户端访问

•本机连接

- 使用初始密码连接

- 修改本机登录密码

- 连接实例

# mysql -uroot -p初始密码 -S sock文件

> alter user user() identified by '新密码';

# mysql -uroot -p新密码 -S sock文件

> show databases;

# ss -antulp | grep mysqld

二、MHA高可用集群

1.搭建环境

1）免密登录

2）主从同步，半同步复制

3）MySQL服务配置

[mysqld]

user=mysql

# mysqld &

# mysql -u -p -S /var/lib/mysql/mysql.sock

> set password for root@localhost = password("123456");

2.部署MySQL高可用集群（MHA软件+主从同步）

集群：使用多台服务器提供相同的服务

高可用集群 主备模式 当主角色的主机宕机后，备用主机自动接替主角的的主机提供服务给客户端。

1）准备MHA运行环境

一主多从

安装依赖的软件包

ssh root 用户无密码登录

2）配置MHA

配置数据节点（一主多从 安装依赖的软件包 彼此之间可以ssh root用户无密码登录）

配置管理主机 server：

装包

修改配置文件

测试配置文件

启动服务

测试高可用集群配置

3.MHA简介

·由日本DeNA公司youshimaton（现就职于Facebook公司开发）

·是一套优秀的座位MySQL高可用性环境下故障切换和主从提升的高可用软件

·目前在MySQL高可用方面是一个相对成熟的解决方案

·在MySQL鼓掌切换过程中，MHA能做到在0~30秒之内自动完成数据库的故障切换操作

·并且在进行故障切换的过程中，MHA能在最大程度上保证数据的一致性，以达到真正意义上的高可用

4.MHA组成

·MHA Manager（管理节点）

- 可以单独在一台独立的机器上管理多个master-slave集群，也可以部署在一台slave节点上

·MHA Node（数据节点）

- 运行在每台MySQL服务器上

·安装MHA

# tar -zxf mha4mysql-manager-0.56.tar.gz

# cd mha4mysql-manager-0.56

# perl Makefile.pl

# make && make install

·修改配置文件

- 彼此之间可以ssh root用户无密码登录

- 配置51时，先开启binlog日志，再授权repluser用户，让52，53一起同步

- 52和53需要作为备用主库需要开启binlog日志

> set globle relay\_log\_purge=off; //所有数据节点都设置，不自动删除本机的中继日志文件

mha4mysql-manager-0.56]# cd samples

# mkdir /etc/mha\_manager/

# cd mha4mysql-manager-0.56/samples/conf/

# cp app1.cnf /etcmha\_manager/

# vim /etc/mha\_manager/app1.cnf

:set nu

[server default]

manager\_workdir=/etc/mha\_manager

manager\_log=/etc/mha\_manager/manager.log

#指定故障迁移脚本名称和路径，自动failover时候的切换脚本

############################

#将原本的脚本拷贝到指定目录下

#脚本原路径mha4mysql-manager-0.56/samples/script/master\_ip\_failover

# cp /root/mha4mysql-manager-0.56/samples/script/master\_ip\_failover /etc/mha\_manager/master\_ip\_failover

############################

master\_ip\_failover\_script=/etc/mha\_manager/master\_ip\_failover

#ssh用户名和端口号

ssh\_user=root

ssh\_port =22

#主从同步用户名和密码

repl\_user=repluser

repl\_password=123456

#连接数据库服务器用户名

user=root

password=123456

[serverX]

#主机

hostname=192.168.4.5X

#竞选主库（1为开启）

candidate\_master=1

#不竞选主库（1为开启）

no\_master=1

·指定命令所在路径

# cd mha4mysql-manager-0.56

# ls bin/

# ls /root/bin

# mkdir /root/bin

# ls /root/bin

# cp bin/\* /root/bin/

# ls /root/bin/

·测试配置文件

> select user,host from mysql.user;

# vim /etc/mha\_manager/app1.cnf

#master\_ip\_failover\_script=/etc/mha\_manager/master\_ip\_failover

# masterha\_check\_ssh --conf=/etc/mha\_manager/app1.cnf

All SSH connection tests passed successfully.

# masterha\_check\_repl --conf=/etc/mha\_manager/app1.cnf

MYSQL Replication Health is OK.

·启动服务

- 查看VIP地址

# ip addr show | grep vip地址

- 把vip地址手动绑定在当前的主库上

# ifconfig eth0 192.168.4.100/24

# ifconfig eth0:1

- 启动服务

# vim /etc/mha\_manager/app1.cnf

master\_ip\_failover\_script=/etc/mha\_manager/master\_ip\_failover

# masterha\_manager --conf=/etc/mha\_manager/app1.cnf

--remove\_dead\_master\_conf //在app1.cnf文件里删除宕机的主库信息

--ignore\_last\_failover //忽略.health文件，否则八小时内主库连续宕机会导致不切换主库

·测试高可用集群配置

- 在数据库服务器上添加访问数据时连接用户 webuser

host51]# mysql

> create database db13;

> grant all on db13.\* to webuser@"%"

-> identified by "123456";

host56]# mysql

> create table db13.a(id int);

-> insert into ...

- 客户端连接VIP地址访问数据库

# mysql -h192.168.4.100 -uwebuser -p123456

- 测试高可用集群

·把主机51上的mysqld停止

·把宕机的数据库服务器51再次加入当前集群中

> change masterto master\_host="192.168.4.52",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master52.00001",

-> master\_log\_pos=154;

> start slave;

# vim /etc/mha\_manager/app1.cnf

[server1]

candidate\_master=1

hostname=192.168.4.51

# mastermha\_check\_repl --conf=/etc/mha\_manager/app1.cnf

> grant all on db12.\* to yaya@"%" identified by "123456";

> grant all on db12.\* to lucy@"192.168.4.%" identified by "123456";

# mysql -h192.168.4.100 -P4006 -ulucy -p123456

# mysql -h192.168.4.100 -P4006 -ulucy -p123456

my $vip = '192.168.4.100/24'; #Virtual IP

my $key = '1';

my #ssh\_start\_vip = "/sbin/ifconfig eth0:$key $vip";

my $ssh\_stop\_vip = "/sbin.ifconfig eth0:$key down";

# vim master\_ip\_failover

# rm -rf /etc/mha\_manager/master\_ip\_failover

# cp /root/mha-soft-student/master\_ip\_failover /etc/mha\_manager/

# vim /etc/mha\_manager/master\_ip\_failover

> select @@hostname

# ifconfig eth0:1 192.168.4.100/24

# mysql -h192.168.4.100 -uwebuser -p123456