

概述

Mysql主从介绍

主从复制步骤

下载地址

解压

安装依赖包

创建用户和组

添加环境变量

建立数据存放目录

初始化数据库

编写配置文件

配置服务启动脚本

启动mysql

修改密码

mysql主从配置

备份主库

解除主库的锁表状态

创建同步账户

配置主数据库编辑配置文件

配置从数据库

启动slave同步进程

测试

常见优化

Too many connections

参考文章汇总

概述

本文基于centos7 mysql 5.7，两台机器，一主一从，搭建最为常见的mysql 主从服务。

主：10.0.221.51 root admin123

从：10.0.221.55

Mysql主从介绍

Mysql主从又叫Replication、AB复制。简单讲就是A与B两台机器做主从后，在A上写数据，另外一台B也会跟着写数据，实现数据实时同步

mysql主从是基于binlog，主上需开启binlog才能进行主从

主从过程大概有3个步骤

主将更改操作记录到binlog里

从将主的binlog事件（sql语句）同步本机上并记录在relaylog里

从根据relaylog里面的sql语句按顺序执行

mysql主从是异步复制过程

master开启bin-log功能，日志文件用于记录数据库的读写增删

需要开启3个线程，master IO线程，slave开启 IO线程 SQL线程，

Slave 通过IO线程连接master，并且请求某个bin-log， position之后的内容。

MASTER服务器收到slave IO线程发来的日志请求信息，io线程去将bin-log内容， position返回给slave IO线程。

slave服务器收到bin-log日志内容，将bin-log日志内容写入relay-log中继日志，创建一个master.info的文件，该文件记录了master ip 用户名 密码 master bin-log名称， bin-log position。

slave端开启SQL线程，实时监控relay-log日志内容是否有更新，解析文件中的SQL语句，在slave数据库中去执行。

主从作用

实时灾备，用于故障切换

读写分离，提供查询服务

备份，避免影响业务

主从形式

一主一从

一主多从（扩展系统读取的性能，因为读是在从库读取的）

多主一从（5.7之后开始）

主主复制

联机复制

主从复制步骤

主库将所有的写操作记录在binlog日志中，并生成log dump线程，将binlog日志传给从库的I/O线程

从库生成两个线程，一个是I/O线程，另一个是SQL线程

I/O线程去请求主库的binlog日志，并将binlog日志中的文件写入relay log（中继日志）中

SQL线程会读取relay log中的内容，并解析成具体的操作，来实现主从的操作一致，达到最终数据一致的目的

主从复制配置步骤

确保从数据库与主数据库里的数据一致

在主数据库里创建一个同步账户授权给从数据库使用

配合主数据库（修改配置文件）

配置从数据库（修改配置文件）

需求

搭建两台MYSQL服务器，一台作为主服务器，一台作为从服务器，主服务器进行写操作，从服务器进行读操作

环境说明

数据库角色 IP 应用与系统 有无数据

主数据库 10.0.221.51 centos7 mysql-5.7 有

从数据库 10.0.221.55 centos7 mysql-5.7 无

下载地址

wget https://cdn.mysql.com/archives/mysql-5.7/mysql-5.7.32-linux-glibc2.12-x86_64.tar.gz

解压

```
tar -zxvf mysql-5.7.32-linux-glibc2.12-x86_64.tar.gz  
  
mv mysql-5.7.32-linux-glibc2.12-x86_64 /usr/local/  
  
cd /usr/local/  
  
ln -sv mysql-5.7.32-linux-glibc2.12-x86_64 mysql
```

安装依赖包

```
yum -y install ncurses-devel openssl-devel openssl cmake mariadb-devel
```

创建用户和组

```
groupadd -r -g 306 mysql
useradd -M -s /sbin/nologin -g 306 -u 306 mysql

chown -R mysql:mysql /usr/local/mysql

ll /usr/local/mysql -d
# lrwxrwxrwx 1 mysql mysql 35 Feb 24 18:03 /usr/local/mysql -> mysql-5.7.32-linux-
glibc2.12-x86_64
```

添加环境变量

```
echo 'export PATH=/usr/local/mysql/bin:$PATH' > /etc/profile.d/mysql.sh
```

如果mysql命令不生效，有可能就是这里导致的

```
chmod 777 /etc/profile.d/mysql.sh
./etc/profile.d/mysql.sh
```

```
#也直接执行
export PATH=/usr/local/mysql/bin:$PATH
```

建立数据存放目录

```
[root@localhost local]# cd /usr/local/mysql
[root@localhost mysql]# mkdir -p /opt/data
[root@localhost mysql]# chown -R mysql:mysql /opt/data/
[root@localhost mysql]# ll /opt/
```

mysql数据存放目录：/opt/data

• 初始化数据库

```
[root@yanyinglai mysql]# /usr/local/mysql/bin/mysqld --initialize --user=mysql --datadir=/opt/data/
```

//这个命令的最后会生成一个临时密码，后面修改初始密码需要

```
# 2021-05-31T12:35:16.661574Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: a7448bff-c20c-11eb-828a-8cec4b6d3911.
#2021-05-31T12:35:16.664229Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
#2021-05-31T12:35:17.118423Z 0 [Warning] CA certificate ca.pem is self signed.
#2021-05-31T12:35:17.563298Z 1 [Note] A temporary password is generated for root@localhost: 7rhfePdje%jP
```

#主库51

```
2021-03-08T03:12:51.489864Z 0 [Warning] CA certificate ca.pem is self signed.
```

```
2021-03-08T03:12:51.587283Z 1 [Note] A temporary password is generated for root@localhost: hsoF
```

pmh-G7FC

#备库55

```
2021-03-08T06:08:47.832562Z 0 [Warning] CA certificate ca.pem is self signed.
```

```
2021-03-08T06:08:48.081076Z 1 [Note] A temporary password is generated for root@localhost: 2tl4,P%fCtbJ
```

编写配置文件

vim /etc/my.cnf

```
[mysqld]
basedir = /usr/local/mysql
datadir = /opt/data
socket = /tmp/mysql.sock
port = 3306
pid-file = /opt/data/mysql.pid
user = mysql
skip-name-resolve
```

配置服务启动脚本

```
[root@localhost mysql]# cp -a /usr/local/mysql/support-files/mysql.server
/etc/init.d/mysqld
[root@localhost mysql]# sed -ri 's#^(basedir=).*#\1/usr/local/mysql#g'
/etc/init.d/mysqld
[root@localhost mysql]# sed -ri 's#^(datadir=).*#\1/opt/data#g' /etc/init.d/mysqld
```

启动mysql

```
[root@localhost mysql]# service mysqld start
Starting MySQL.Logging to '/opt/data/localhost.localdomain.err'.
SUCCESS!
```

```
service mysqld restart #重新启动mysql
```

```
[root@localhost mysql]# ss -antl
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	10.0.221.51:7001	*:*

```
LISTEN      0      100      127.0.0.1:25      *:*

LISTEN      0      128             *:10050           *:*

LISTEN      0      128      10.0.221.51:17001 *:*

LISTEN      0      128             *:22              *:*

LISTEN      0      100             :::1:25           :::*

LISTEN      0      80             :::3306           :::*

LISTEN      0      128             :::22             :::*
```

修改密码

使用临时密码修改

```
[root@yanyinglai ~]# mysql -uroot -p
mysql> set password = password('admin123');
mysql> quit
```

修改root用户权限为所有机器均可登录

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@ '%' IDENTIFIED BY 'admin123' WITH GRANT
OPTION;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

mysql主从配置

确保从数据库与主数据库的数据一样先在主数据库创建所需要同步的库和表

```
[root@localhost mysql]# mysql -uroot -padmin123
```

```
mysql> create database yan;
Query OK, 1 row affected (0.00 sec)

mysql> create database lisi;
Query OK, 1 row affected (0.00 sec)

mysql> create database wangwu;
Query OK, 1 row affected (0.00 sec)

mysql> use yan;
Database changed
mysql> create table tom (id int not null,name varchar(100)not null ,age tinyint);
Query OK, 0 rows affected (11.83 sec)

mysql> insert tom (id,name,age) values(1,'zhangshan',20),(2,'wangwu',7),(3,'lisi',23);
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from tom;
```

备份主库

备份主库时需要另开一个终端，给数据库上读锁，避免在备份期间有其他人在写入导致数据同步的不一致

```
[root@localhost mysql]# mysql -uroot -padmin123
```

```
mysql> flush tables with read lock;
Query OK, 0 rows affected (0.01 sec)
```

//此锁表的终端必须在备份完成以后才能退出（退出锁表失效）

备份主库并将备份文件传送到从库


```
[root@rm-redis-2 /]# mysqldump -uroot -padmin123 --all-databases > /opt/all-20210308.sql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

```
[root@localhost ~]# scp /opt/all-20210308.sql root@10.0.221.55:/opt/
```

解除主库的锁表状态

解除主库的锁表状态，直接退出交互式界面即可

```
mysql> quit
Bye
```

在从库上恢复主库的备份并查看是否与主库的数据保持一致

```
mysql -uroot -padmin123 < /opt/all-20210308.sql
```

创建同步账户

在主数据库创建一个同步账户授权给从数据使用 IP对应为slave的

```
mysql -uroot -padmin123

create user 'repl'@'10.0.221.55' identified by 'a123456';

grant replication slave on *.* to 'repl'@'10.0.221.55' ;

flush privileges ;
```

如果创建失误，可以先删除，再新建，命令如下：

```
use mysql;
delete from user where User='repl';
```

```
mysql -uroot -padmin123

create user 'repl'@'%' identified by 'a123456';

grant replication slave on *.* to 'repl'@'%' ;

flush privileges ;
```

配置主数据库编辑配置文件

```
[root@localhost ~]# vi /etc/my.cnf
```

//添加以下内容

//启用binlog日志

//主数据库服务器唯一标识符 主的必须必从大

```
log-bin=mysql-bin
server-id=1
log-error=/opt/data/mysql.log
```

```
[root@yanyinglai ~]# ss -antl
State Recv-Q Send-Q Local Address:Port Peer Address:Port
LISTEN 0 128 *:22 *:22
LISTEN 0 100 127.0.0.1:25 *:25
LISTEN 0 128 :::22 :::22
LISTEN 0 100 :::1:25 ::::1:25
LISTEN 0 80 :::3306 ::::3306
```

查看主库的状态

```
mysql> show master status;
```

```
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000002 | 154 | | | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

配置从数据库

编辑配置文件

添加以下内容：

```
//设置从库的唯一标识符 从的必须比主小
```

```
//启用中继日志relay log
```

```
server-id=2
relay-log=mysql-relay-bin
error-log=/opt/data/mysql.log
```

[root@rdtest mysql]# vim /etc/my.cnf

```
[mysqld]
basedir = /usr/local/mysql
datadir = /opt/data
socket = /tmp/mysql.sock
port = 3306
pid-file = /opt/data/mysql.pid
user = mysql
skip-name-resolve

server-id=2
relay-log=mysql-relay-bin
log-error=/opt/data/mysql.log
```

重启从库的mysql服务

```
[root@rdtest mysql]# service mysqld restart
Shutting down MySQL.. SUCCESS!
Starting MySQL.Logging to '/opt/data/mysql.log'.
SUCCESS!
```

```
[root@rdtest mysql]# ss -antl
State Recv-Q Send-Q Local Address:Port Peer Address:Port
LISTEN 0 128 :::22 :::*
LISTEN 0 128 *:22 *:~
LISTEN 0 100 :::1:25 :::*
LISTEN 0 100 127.0.0.1:25 *:~
LISTEN 0 10 *:5672 *:~
LISTEN 0 80 :::3306 :::*
[root@rdtest mysql]#
```

启动slave同步进程

在主数据库中通过 `show master status` 命令查看最新的master_log_pos;

```
change master to
master_host='10.0.221.51',
master_user='repl',
master_password='a123456',
master_log_file='mysql-bin.000002',
master_log_pos=3052;
```

查看从服务器状态

```
mysql> show slave status\G;
Slave_IO_Running: Yes //此处必须是yes
Slave_SQL_Running: Yes //此处必须是yes
```

测试

测试验证在主服务器的yan库的tom表插入数据:

```
mysql> use yan;
mysql> select * from tom;
mysql> insert tom(id,name,age) value (4,"zgy",18);
Query OK, 1 row affected (0.14 sec)
mysql> select * from tom;
```

```
insert tom(id,name,age) value (5,"zgy",16);
update tom set name='yyyyy' where age=18;
delete from tom where age=18;
```

在从数据库查看是否数据同步

```
mysql> use yan;  
mysql> select * from tom;
```

关注Slave_IO_State,Slave_IO_Running,Slave_SQL_Running状态

若都为yes状态时确认同步配置完成

常见优化

- Too many connections

出现此类错误时，Data source rejected establishment of connection, message from server: "Too many connections", 是因为默认连接数过小导致的。

Mysql默认的最大连接数是151，可以使用如下命令进行查看：

```
show status like '%connections';
```

在[mysqld] 下面添加下面三行

```
max_connections=1000  
max_user_connections=500  
wait_timeout=200
```

参考文章汇总

[mysql主从简单配置](#)

[MySQL创建用户与授权](#)

[mysql主从配置实现一主一从读写分离](#)