

第25章 MySQL replication(主从)配置

学习Linux请加QQ群： 群1(163262181) 群2(148412746) 群3(246401509) 群4(173884211)

跟阿铭学Linux邀请函 (<http://www.aminglinux.com>)，猿课已上线，请加微信aminglinux84索要配套视频教程。

MySQL Replication 又叫做AB复制或主从复制。它主要用于MySQL的时时备份或者读写分离。在配置之前先做一下准备工作，配置两台mysql服务器，或者在一台服务器上配置两个端口也可以。阿铭本章的实验中就是在在一台服务器上跑了两个mysql。

配置mysql服务

详细步骤，请参考(<http://study.lishiming.net/chapter17.html#mysql>), 阿铭只把简单步骤写一下。

根据阿铭提供的地址，假如你已经搭建好了一个mysql，跑的是3306端口，下面阿铭再搭建一个3307端口的mysql：

```
[root@localhost ~]# cd /usr/local/
[root@localhost local]# cp -r mysql mysql_2
[root@localhost local]# cd mysql_2
[root@localhost mysql_2]# ./scripts/mysql_install_db --user=mysql --datadir=/data/mysql2
```

最后一步是初始化数据库目录，如果出现两个“OK”并且生成/data/mysql2目录才正确，否则请仔细查看错误信息，如果不能解决请到阿铭论坛(<http://www.aminglinux.com/bbs/forum-40-1.html>)发帖咨询阿铭。拷贝配置文件到mysql_2下，并修改相关项目：

```
[root@localhost mysql_2]# cp /etc/my.cnf ./my.cnf
[root@localhost mysql_2]# vim my.cnf
```

其中：

```
port = 3306
```

改为：

```
port = 3307
```

把：

```
socket = /tmp/mysql.sock
```

改为：

```
socket = /tmp/mysql2.sock
```

在这一行的下面再加一行：

```
datadir = /data/mysql2
```

保存后就可以启动它了：

```
[root@localhost mysql_2]# cd bin/
[root@localhost bin]# ./mysqld_safe --defaults-file=../my.cnf --user=mysql &
```

如果以后想开机启动，就把它加入/etc/rc.d/rc.local文件中：

```
echo ". /mysqld_safe --defaults-file=../my.cnf --user=mysql &" >>/etc/rc.d/rc.local
```

到此，目前阿铭已经在Linux上启动了两个mysql：

```
[root@localhost ~]# netstat -lnp |grep mysqld
tcp        0      0 0.0.0.0:3306          0.0.0.0:*        LISTEN     3169/mysqld
tcp        0      0 0.0.0.0:3307          0.0.0.0:*        LISTEN     3037/mysqld
unix  2      [ ACC ]     STREAM  LISTENING   29027  3037/mysqld  /tmp/mysql2.sock
unix  2      [ ACC ]     STREAM  LISTENING   29155  3169/mysqld  /tmp/mysql.sock
```

配置replication

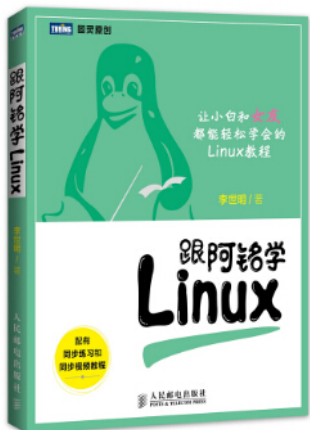
阿铭打算把3307端口的mysql作为主(master)，而把3306的mysql作为从(slave)。为了让实验更加像生产环境，所以阿铭先在master上创建一个库db1，并且把mysql的库数据复制给它：

```
[root@localhost bin]# mysql -uroot -S /tmp/mysql2.sock
```

目录列表

第1章 前言
第2章 关于Linux的历史
第3章 对Linux系统管理员的建议
第4章 安装Linux操作系统
第5章 初步认识Linux
第6章 Linux系统的远程登陆
第7章 Linux文件与目录管理
第8章 Linux系统用户及用户组管理
第9章 Linux磁盘管理
第10章 文本编辑工具vim
第11章 文档的压缩与打包
第12章 安装RPM包或者安装源码包
第13章 学习 shell脚本之前的基础知识
第14章 正则表达式
第15章 shell脚本
第16章 linux系统日常管理
第17章 LAMP环境搭建
第18章 LNMP环境搭建
第19章 学会使用简单的MySQL操作
第20章 NFS服务配置
第21章 配置FTP服务
第22章 配置Squid服务
第23章 配置Tomcat
第24章 配置Samba服务器
第25章 MySQL replication(主从)配置
结语

阿铭著作：



微信扫码获取最新版linux电子书和视频

Enter search terms or a module, class or function name.

```
mysql> create database db1;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> quit
Bye
```

也许你对阿铭使用的这个命令有点疑问, **-S** 后面指定**mysql**的**socket**文件路径, 这也是登陆**mysql**的一种方法, 因为在一台服务器上跑了两个**mysql**端口, 所以, 只能用 **-S** 这样的方法来区分。首先创建了**db1**库, 然后把**mysql**库的数据复制给它:

```
mysqldump -uroot -S /tmp/mysql2.sock mysql > 123.sql
mysql -uroot -S /tmp/mysql2.sock db1 < 123.sql
```

1. 设置master

修改配置文件:

```
vim /usr/local/mysql_2/my.cnf
```

在**[mysqld]**部分查看是否有以下内容, 如果没有则添加:

```
server-id=1
log-bin=mysql-bin
```

除了这两行是必须的外, 还有两个参数, 你可以选择性的使用:

```
binlog-do-db=databasename1,databasename2
binlog-ignore-db=databasename1,databasename2
```

binlog-do-db=需要复制的数据库名, 多个数据库名, 使用逗号分隔。**binlog-ignore-db**=不需要复制的数据库名, 多个数据库名, 使用逗号分隔。这两个参数其实用一个就可以啦。

如果修改过配置文件需要重启**mysqld**服务, 否则不需要重启:

```
[root@localhost ~]# pid=`ps uax |grep mysql2.sock |grep -v grep |awk '{print $2}'`
[root@localhost ~]# kill -0 $pid; sleep 3; kill $pid
[root@localhost ~]# cd /usr/local/mysql_2/bin/
[root@localhost bin]# ./mysqld_safe --defaults-file=../my.cnf --user=mysql &
```

由于阿铭没有编写启动脚本, 所以**3307**端口的**mysql**重启有点麻烦。

设置**mysql**数据库的**root**访问密码:

```
mysqladmin -u root -S /tmp/mysql2.sock password '123456'
mysql -u root -S /tmp/mysql2.sock -p'123456'
```

```
mysql> grant replication slave on *.* to 'repl'@'127.0.0.1' identified by '123123';
//这里的repl是为slave端设置的访问master端mysql数据的用户, 密码为123123, 这里的127.0.0.1为slave的ip (因为阿铭配置的master和slave都在本机)。
mysql> flush tables with read lock; //锁定数据库, 此时不允许更改任何数据
mysql> show master status; //查看状态, 这些数据是要记录的, 一会要在slave端用到
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000006 | 474952 |              |                   |
+-----+-----+-----+-----+
```

2. 设置slave

先修改**slave**的配置文件**my.cnf**:

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

找到 **“server-id = 1”** 这一行, 删除掉或者改为 **“server-id = 2”** 总之不能让这个**id**和**master**一样, 否则会报错。另外在从上, 你也可以选择性的增加如下两行, 对应于主上增加的两行:

```
replicate-do-db=databasename1,databasename2
replicate-ignore-db=databasename1,databasename2
```

改完后, 重启**slave**:

```
service mysqld restart
```

拷贝**master**上的**db1**库的数据到**slave**上, 因为**master**和**slave**都在一台服务器上, 所以操作起来简单了很多, 如果是不同的机器, 可能就需要远程拷贝了, 希望你注意这一点:

```
[root@localhost ~]# mysqldump -uroot -S /tmp/mysql2.sock -p123456 db1 > db1.sql
[root@localhost ~]# mysql -uroot -S /tmp/mysql.sock -pyourpassword -e "create database db1"
[root@localhost ~]# mysql -uroot -S /tmp/mysql.sock -pyourpassword db1 < db1.sql
```

第二行中, 阿铭使用了一个**-e**选项, 这个选项阿铭在之前的章节中并没有介绍, 它用来把**mysql**的命令写到

shell中，这样可以方便把mysql操作写进脚本中，它的格式就是 `-e "command"` 它很实用，阿铭用的也是蛮多的，所以请你熟记它。把数据拷贝过来后，就需要在slave上配置主从了：

```
[root@localhost ~]# mysql -uroot -S /tmp/mysql.sock -pyourpassword
mysql> slave stop;
mysql> change master to master_host='127.0.0.1', master_port=3307,
master_user='repl', master_password='123123',
master_log_file='mysql-bin.000006', master_log_pos=474952;
mysql> slave start;
```

相信聪明的你一定可以看懂上面的各个参数分别表示什么含义，其中`master_log_file`和`master_log_pos`是在上面使用 `show master status` 查到的数据。执行完这一步后，需要在master上执行一步：

```
mysql -uroot -S /tmp/mysql2.sock -p123456 -e "unlock tables"
```

然后查看slave的状态：

```
mysql> show slave status\G;
```

确认以下两项参数都为yes：

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

测试主从

在master上执行如下命令：

```
[root@localhost ~]# mysql -uroot -S /tmp/mysql2.sock -p123456 -e "use db1;
select count(*) from db"
+-----+
| count(*) |
+-----+
|          2 |
+-----+

[root@localhost ~]# mysql -uroot -S /tmp/mysql2.sock -p123456 -e "use db1;
truncate table db"

[root@localhost ~]# mysql -uroot -S /tmp/mysql2.sock -p123456 -e "use db1;
select count(*) from db"
+-----+
| count(*) |
+-----+
|          0 |
+-----+
```

这样清空了db1.db表的数据，下面查看slave上的该表数据：

```
[root@localhost ~]# mysql -uroot -S /tmp/mysql.sock -pyourpassword -e "use db1; select count(*) from db"
+-----+
| count(*) |
+-----+
|          0 |
+-----+
```

slave上的该表也被清空了。这样好像不太明显，不妨继续把db表删除试试：

```
[root@localhost ~]# mysql -uroot -S /tmp/mysql2.sock -p123456 -e "use db1; drop table db"
[root@localhost ~]# mysql -uroot -S /tmp/mysql.sock -pyourpassword -e "use db1; select count(*) from db"
ERROR 1146 (42S02) at line 1: Table 'db1.db' doesn't exist
```

这次很明显了。

主从配置起来很简单，但是这种机制也是非常脆弱的，一旦我们不小心在从上写了数据，那么主从也就被破坏了。另外如果重启master，务必要先把slave停掉，也就是说需要在slave上去执行 `slave stop` 命令，然后再去重启master的mysql服务，否则很有可能就会中断了。当然重启完后，还需要把slave给开启 `slave start`。

阿铭建议你最好再扩展学习一下：<http://www.aminglinux.com/bbs/thread-5449-1-1.html>

教程答疑：[请移步这里](#)。

欢迎你加入 [阿铭学院](#) 和阿铭一起学习Linux，让阿铭成为你Linux生涯中永远的朋友吧！