

叮价格斐呢

博客园 首页 新随笔 联系 订阅 管理

mysql的主主复制详解

Mysql双主部署

解释：

所谓双主备份，其实也就是互做主从复制，每台master既是master，又是另一台服务器的slave。这样，任何一方所做的变更，都会通过复制应用到另外一方的数据库中。

要求：部署Mysql双主复制，部署期间生产服务器可正常访问，但无法进行登录/注册/退出等涉及到对Mysql写的操作。

环境：

系统：centos 6.7

mysql版本：5.6.28

生产环境共2台mysql服务器（以下对两台mysql分别简称为A B）

A：192.168.1.1 （master）

B：192.168.1.2 （slave）

部署前的准备

分别检查A、B两台Mysql服务器是否作为其他master的slave，并停止复制，以免受到影响。

#show slave status\G;

#mysql>stop slave;

. master (A库) ---> slave (B库)

开始部署

1. 同步数据：以A库的数据为准，同步A B数据，使AB数据保持一致。

1.1 A中导出数据

（1）对A库加锁，避免同步时数据发生改变:

mysql>use database_name;

mysql> flush tables with read lock; #关闭所有打开的表，同时对于所有数据库中的表都加一个读锁，直到显示地执行unlock tables，该操作常常用于数据备份的时候。也就是将所有的脏页都要刷新到磁盘，然后对所有的表加上了读锁，于是这时候直接拷贝数据文件也就是安全的。为此库下的所有表施加读锁，退出会话后失效，所谓读锁，也就是只能读，不能写。

（2）使用mysqldump工具导出数据:

mysqldump -uroot -pxxx -R database_name >database_name.sql # -R参数表示同时备份函数

公告

昵称：叮价格斐呢
园龄：1年3个月
粉丝：6
关注：4
+加关注

< 2019年3			
日	一	二	三
24	25	26	27
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27
31	1	2	3

搜索

我的标签

linux基础(32)

mysql(28)

python(13)

linux杂谈(11)

shell(10)

Linux主机安全(7)

tomcat(6)

VPN(6)

(3) 记下binlog日志中Position的位置（用于slave连接A做复制时会用到）

```
mysql> show master status\G;          # mysql-bin.000003 pos=443
```

(4) 解锁数据库:

```
mysql>unlock tables;
```

1.2 将A备份的数据导入到B中

```
[root@localhost ~]# mysql -uroot -pxxx < database_name.sql
```

2. 配置Master (A)

(1) master中创建一个仅用于主从复制的账号，给与REPLICATION SLAVE权限。此权限仅允许slave访问Master的bin-log日志。

在Master中创建一个仅用于复制的mysql用户，并仅给与复制权限。

```
mysql> create user 'repl_A'@'192.168.1.2' identified by '123';
```

```
mysql> grant replication slave on *.* to 'repl_A'@'192.168.1.2';
```

或

```
mysql>grant replication slave on *.* to 'repl_A'@ '192.168.1.2' identified by '123';  #上面两句可以直接简略为这一句。
```

(2) master开启二进制日志

```
vim /etc/my.cnf 或者 vim /usr/my.cnf （在[mysqld]下添加：）
```

```
server-id=1          #指定master主机的id，不可为0，否则拒绝所有slave连接。
```

```
log-bin=mysql_bin    #指定bin-log文件前缀名称，开启binlog日志
```

```
binlog_do_db=database_name    #指定binlog日志是记录的是哪个库
```

```
replicate-do-db=database_name    #指定复制哪一个库
```

```
auto-increment-increment = 2  //每次增长2
```

```
auto-increment-offset = 1  //设置自动增长的字段的偏移量，即初始值为1
```

```
log_bin_trust_function_creators=1      #详解如下方
```

```
expire_logs_days = 10      #保留10天的bin_log日志，防止日志太多占用磁盘空间
```

```
max_binlog_size = 100M      #限制每个bin_log日志大小最大为100M。
```

```
log-slave-updates=1      #slave执行master的sql后，将sql记录在binlog日志中（默认是不记录的）——实际生产我没开启这条
```

```
max_connections=600      #指定最大连接数
```

```
wait_timeout=5          #等待超时
```

#重启mysql使配置生效。

3. 配置Slave (B)

(1) 配置slave的server-id

示例：[root@iz2z usr]# vim my.cnf

```
[mysqld]
```

```
server-id=2
```

```
auto-increment-increment = 2  //每次增长2
```

```
auto-increment-offset = 2  //设置自动增长的字段的偏移量，即初始值为2
```

注：二都只有server_id不同和 auto-increment- offset不同

auto-increment-increment的值应设为整个结构中服务器的总数，本案例用到两台服务器，所以值设为2。

重启mysql使配置生效。

服务器安全(6)
阿里云(5)
更多

随笔档案
2019年2月 (8)
2019年1月 (4)
2018年11月 (8)
2018年10月 (9)
2018年9月 (4)
2018年8月 (9)
2018年7月 (2)
2018年6月 (6)
2018年5月 (3)
2018年4月 (7)
2018年3月 (10)
2018年2月 (1)
2018年1月 (21)
2017年12月 (35)

文章档案
2018年3月 (5)
2018年1月 (3)
2017年12月 (48)
2017年11月 (3)

最新评论
1. Re:jenkins和jdk版本的确，jenkins自身运行任务
的确，jenkins自身运行任务
的jdk，是独立的。

（2）配置连接master的相关信息配置。

在slave中指定master相关参数。

你可以一行一行的配置master连接信息。

```
mysql>change master to master_host='192.168.1.1', #指定master主机IP地址
>master_user='repl_A', #指定一个（在master中）有复制权限的用户
>master_port=3306,
>master_password='123', #指定上述有复制权限用户的密码
>master_log_file='mysql-bin.000003', #指定开始复制的binlog二进制文件。
>master_log_pos=443; #指定（在master二进制文件中）要开始复制的位置。#注意： master_log_pos的值无需加引号，否则报错！
```

或者

将master连接信息配置写在一行。

```
mysql>change master to master_host='192.168.1.1',master_user='repl',master_password='123',master_log_file='mysql-
bin.000003',master_log_pos=443;
```

解释：

连接master的配置信息在命令行中配置好后，默认存在/var/lib/mysql/master.info文件，可以使用show master status\G;命令查看，所有就算重启也不用担心连接master的配置丢失。

（3）开启slave

slave中：start slave；

Slave_IO_Running和Slave_SQL_Running的状态都是YES,说明同步开启成功。

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

二. master（B库） ---> slave（A库）

Master(B)配置

（1）master中创建一个仅用于主从复制的账号

```
mysql> grant REPLICATION SLAVE on *.* to 'repl_B'@'192.168.1.1' identified by '123';
```

（2）查看B库中binlog日志的位置

```
mysql> show master status\G; # mysql_bin.000004 pos=558
```

（3）同步数据

```
mysql>change master to master_host='192.168.1.2',
>master_user='rel_B,
>master_port=3306,
>master_password='123',
>master_log_file='mysql_bin.000004',
>master_log_pos=558;
```

（4）开启slave

A（slave）库中：start slave；

阅读排行榜

1. 自动化运维python学

2. mysql高可用架构之M
现读写分离详解(2418)

3. mysql的主主复制详解

4. 记一次阿里云服务器被
肉鸡(1877)

5. linux 下端口close_w

评论排行榜

1. jenkins和jdk版本问题

推荐排行榜

1. linux系统安全审计简

2. HAProxy+keepalived
A中slave集群负载均衡

3. 自动化运维python学

4. jenkins和jdk版本问题

(5) 测试验证

解释:

log_bin_trust_function_creators=1

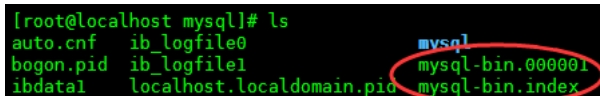
当二进制日志启用后, 这个变量就会启用。它控制是否可以信任存储函数创建者, 不会创建写入二进制日志引起不安全事件的存储函数。如果设置为0 (默认值), 用户不得创建或修改存储函数, 除非它们具有除CREATE ROUTINE或ALTER ROUTINE特权之外的SUPER权限。如果变量设置为1, MySQL不会对创建存储函数实施这些限制。此变量也适用于触发器的创建。

server-id=1

原理: 在master中, 您必须配置一个独特的服务器ID (可能需要重新启动服务器) 和启用二进制日志, slave的复制基于master的二进制日志。如果没有使用log-bin选项启用二进制日志, 复制是不可能的。每个slave与master必须配置一个独特的ID。这个ID用于识别主从服务器, 而且必须是一个正整数1和(232)-1之间。

验证:

1. ls /var/lib/mysql/ 会发现有mysql-bin.000001mysql、mysql-bin.index, 表示启用二进制日志成功。



```
[root@localhost mysql]# ls
auto.cnf      ib_logfile0
bogon.pid     ib_logfile1
ibdata1       localhost.localdomain.pid
mysql-bin.000001
mysql-bin.index
```

mysql_bin.000001: Master记录操作的二进制日志文件。

mysql_bin.index: 二进制文件的索引文件, 存放二进制日志文件的名字和路径。

2. 命令查看

#查看是否开启了bin-log mysql> show variables like '%log_bin%';

#查看bin-log相关参数 mysql> show variables like '%binlog%';

#当前mysql二进制日志文件的默认位置 mysql> show variables like '%datadir%';

#查看mysql二进制文件名称及大小 mysql> show binary logs;

3. Master常用命令

#查看Master复制状态 mysql> show master status;

#查看二进制日志文件事件内容 mysql> show binlog events in 'mysql-bin.000001';

#查看二进制日志文件事件内容 mysql> show binlog events in 'mysql-bin.000003';

4. 管理二进制日志文件

#删除二进制日志文件

手动删除:

语法: > PURGE {MASTER | BINARY} LOGS TO 'log_name'

> PURGE {MASTER | BINARY} LOGS BEFORE 'date'

#我使用时没发现MASTER 和BINARY的区别, 都一样用, 都可以。

mysql> purge binary logs to 'mysql_bin.000002';

//删除mysql_bin.000002之前的二进制日志文件, 并同步删除日志索引文件mysql_bin.index中mysql_bin.000002之前的索引信息。注意不包含mysql_bin.000002

mysql> purge master logs before '2017-07-20 21:58:00'; //清除2017-07-20 21:58:00 之前的二进制日志, 并同步清楚日志索引文件的信息。

自动删除:

my.cnf中加入以下, 然后重启:

SET GLOBAL expire_logs_days = 7 \\设置二进制日志只保留7天的

-----如果不想重启直接mysql> SET GLOBAL expire_logs_days = 7

#重置master二进制日志文件及内容

mysql> reset master;

或者是

mysql>flush logs;

最难不过坚持

标签: mysql

好文要顶

关注我

收藏该文







叮你格斐呢

关注 - 4

粉丝 - 6

+加关注

0

0

« 上一篇： 下载系统已经安装的rpm包

» 下一篇： SVN上线步骤笔记

posted @ 2017-12-18 17:23 叮你格斐呢 阅读(2093) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】超50万C++/C#源码: 大型实时仿真HMI组态CAD\GIS图形源码！
- 【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

相关博文：

- mysql主从复制 详解
- MySQL主从复制详解
- 详解MySQL主从复制实战 - 基于GTID的复制
- Mysql主从复制原理详解
- mysql主从复制安装详解

最新新闻：

- 滴滴试行“选择路线”功能 为乘客提供至多三条不同线路
- 机器人黄金时代即将到来？协作机器人渐成主流
- 对话刘作虎：因为我们的用户很极客，所以我们要做 5G 手机
- 雷军两会提案：推动5G与物联网融合 大力发展商业航天
- 为了发力服务业务，苹果招聘的软件工程师数量已经超过了硬件
- » 更多新闻...

