

MHA不作为重点，出现已经很早了。有点更不上了，作者已经不在维护了。

首先是安装



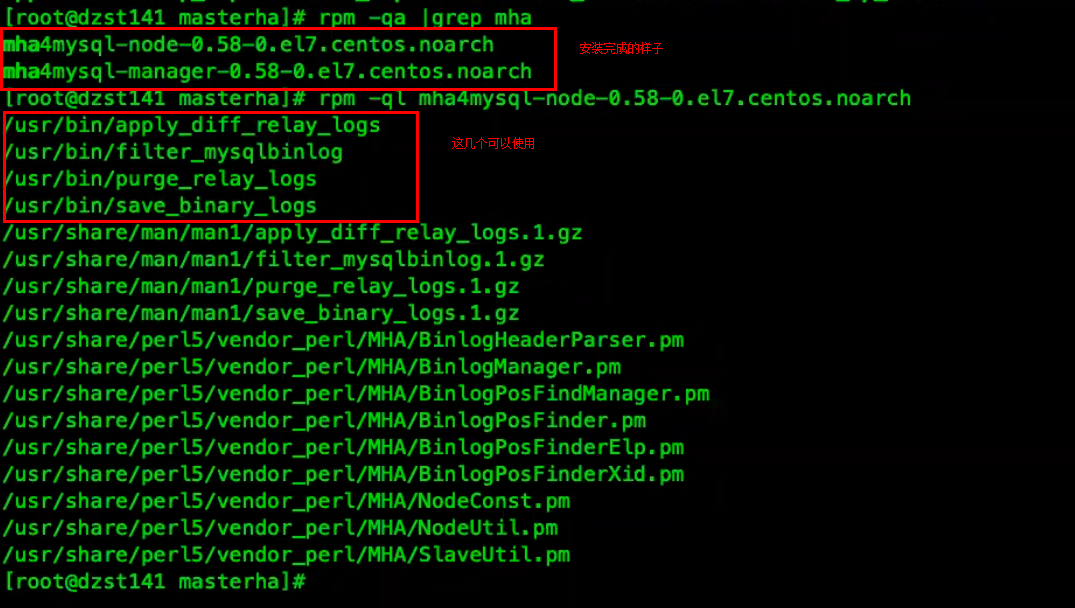
Centos7也可以使用，但是依赖包很多。最好安装上centos的epel源，之后再来安装。

安装来找百度文档来安装。



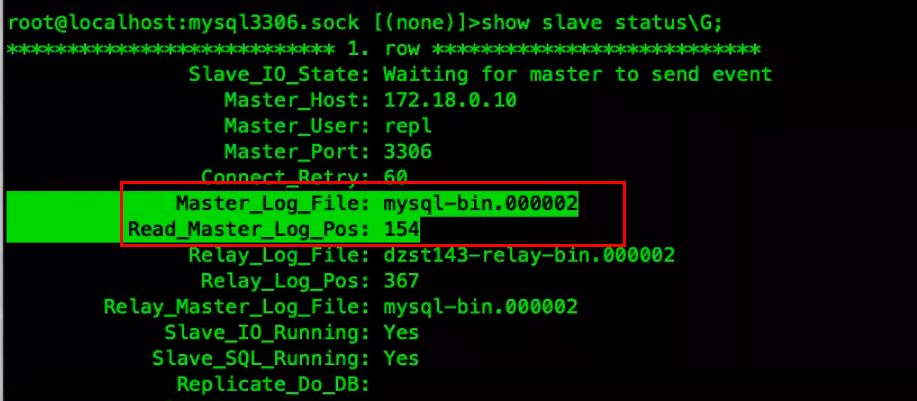
安装MHA时依赖的包

安装完的样子，可以看到一个是node，一个是manager



Save\_binary\_logs命令可能是我们唯一要用到的，作用是什么呢？

我们show slave status\G;可以看到



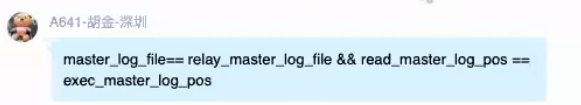
这两个参数，那么出现场景一：

Master挂了或者slave挂了之后怎么判断日志的差别？也就是看同步是不是完毕了，要有科学的方法（使用程序来帮助我们校验）。

1 首先要对比master\_last\_file 和 read\_last\_file是不是一样，如果一样进行第二个步骤

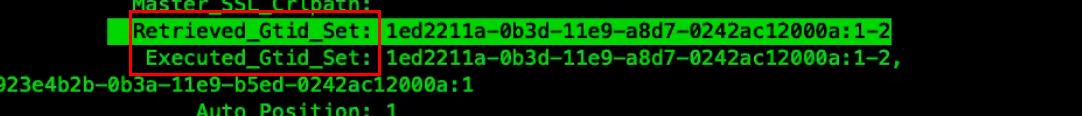
2 如果master\_last\_file和relay\_master\_log\_file 是一样的情况下，对比pos（也就是位置）是不是一样

如果file文件一样，pos一样，主从的日志就是一样的了。



有GTID环境：

执行了show master status；可以获取到

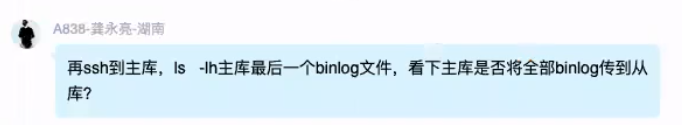
想方设法（可以使用脚本）对比这两个值是不是一样，就可以判断出来，**可以百度一下这两个参数是什么意思。**如果一样就可以判断io\_thread获取的日志被sql\_thread重放完毕。



Read\_master\_log\_pos – exec\_master\_log\_pos大于0 就表示有延迟，单位是字节。

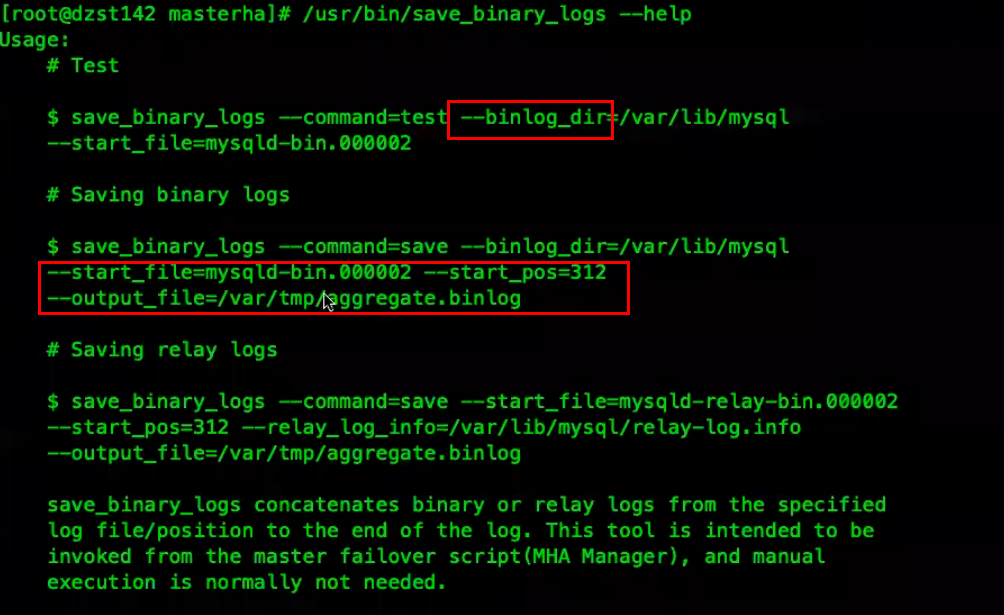
场景二：

如果说主库挂掉了，怎么判断主库有没有把日志全部传输给从库呢？



这是一个解决方案。

MHA也做这个，save\_binary\_logs 就是



查看参数start\_file , start\_pos , output\_file 这三个参数表示从那个mysqld\_bin文件的那个pos开始，保存到output\_file指定的文件。

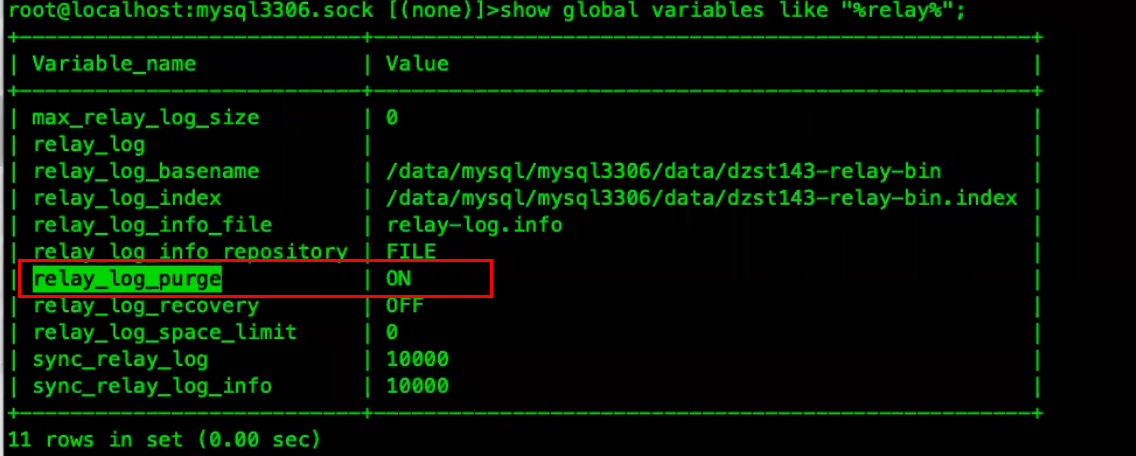
所以save\_binary\_logs的作用就是把从库的开始文件，开始位置记录下来，把这个文件及文件位置之后的主库上的binlog文件输出到output\_file文件里面，再把这个文件传递到从库上，从库进行重放达到数据一致。

理解：就是从库把自己最后获取的binlog文件及其binlog文件的pos传递给save\_binary\_logs（MHA），然后save\_binary\_logs（MHA）根据从库传递过来的binlog文件及其位置，把主库上之后的binlog从定向到一个文件（新的binlog），再把这个binlog传递给从库，从库通过重放binlog达到数据一致。

那么思考？增强半同步还是需要这个逻辑么？

场景二：

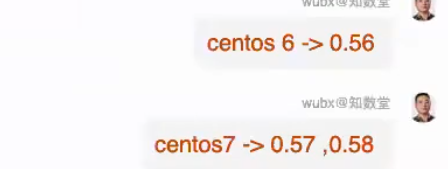
Master的主库也挂掉了，无法ssh登陆上去。

这时候可能出现的情况就是s1比s2，s3同步的日志都多，这种情况下利用s1把s2和s3的数据补全，这个时候需要把relay\_log\_purge这个参数关闭掉，关闭掉意味着主库上的relay

再应用完之后也不会被删除，然后补全即可。

但是如果是GTID环境就不需要这样操作了，直接change就可以了。

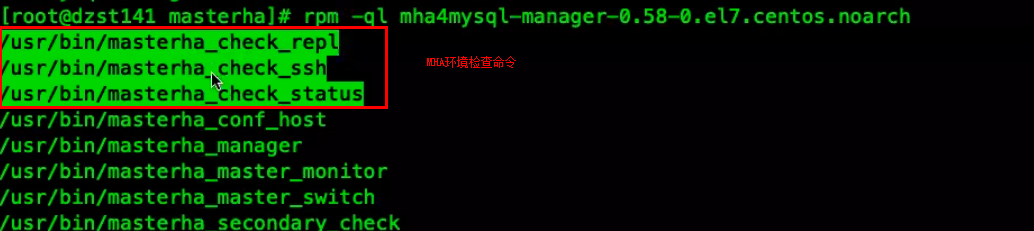
MHA和操作系统的对应关系：



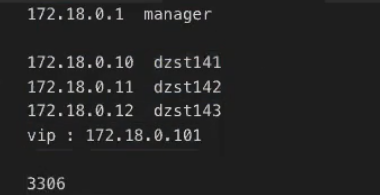
MHA的难点是安装。

主机链接不上可以通过binlog server来补全日志。

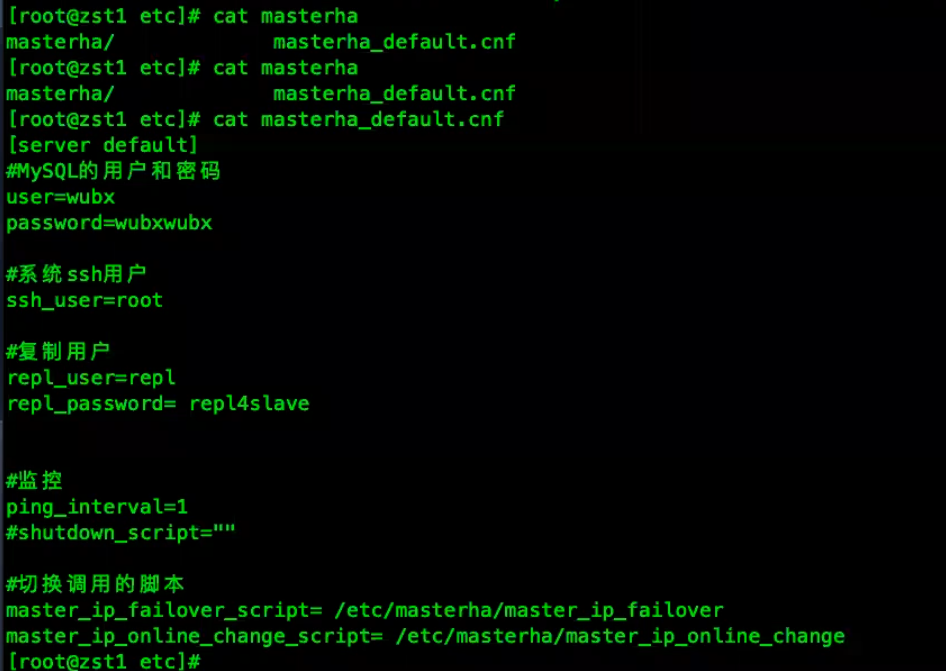
-------------node------------



老师的测试环境



这几个机器需要ssh信任。



登陆用的账号权限：select ，mysql.user , set global super\_read\_only=1; , super 权限。

使用MHA最好重写这两个脚本

