oneproxy切库切表

oneprox：垂直分布，水平分表，proxy集群，读高可用，读写分离（master不参与读取），读写分离（master参与读取），写高可用，读写随机

垂直切库：降低读取压力

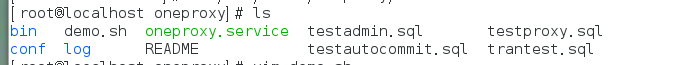
水平切表：降低写入压力

实验环境：三台mysql环境，其中一台安装oneproxy，其他两台作为存放数据的数据库

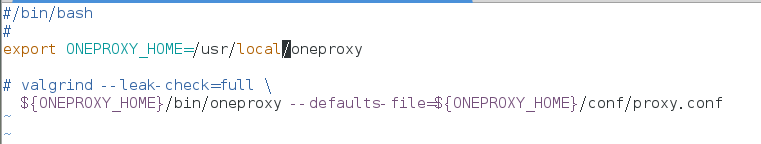


LNMP作为oneproxy；数据库192.168.10.1作为主服务器；数据库192.168.10.4作为从服务器

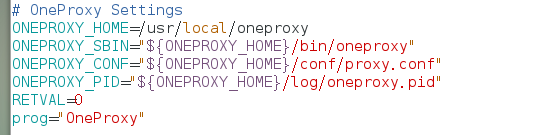
解压包，并在家目录下查看到该内容



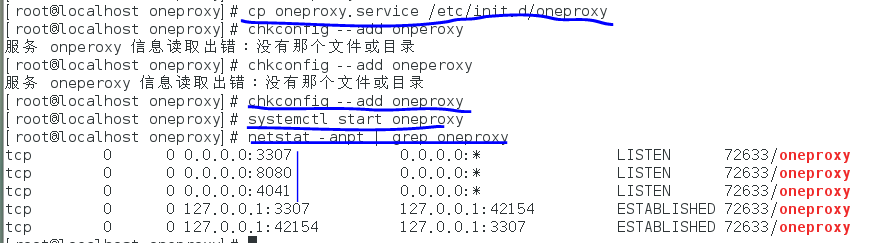
修改demo.sh文件，将目录修改为家目录



修改oneproxy.service中的目录为家目录



复制onerous.service复制到/etc/init.d/下



在另外两台做主从复制

修改主服务器中的my.cnf文件中server-id=1，log-bin=mysql-bin（5.6版本添加，5.5版本可以不添加）



修改从服务器中的my.cnf文件中server-id=2

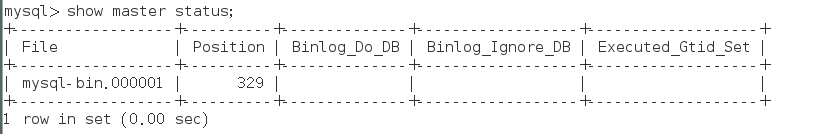


重启mysql

主服务器上授权

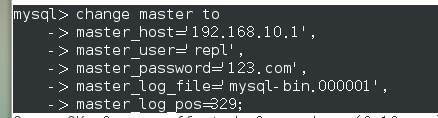
grant replication slave on \*.\* to 'repl'@'192.168.10.%' identified by '123.com';

查看节点：

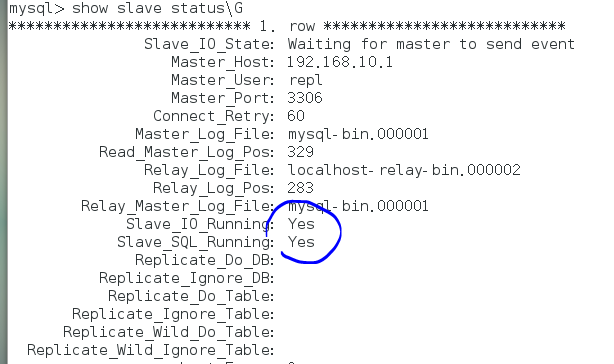


从服务器上：

change master to master\_host='192.168.10.1',master\_user='repl',master\_password='123.com',master\_log\_file='mysql-bin.000001', master\_log\_pos=329;



show slave status\G查看

若第一次不成功，检查防火墙关闭且能ping通的情况下，通过stop slave关闭后start slave启动在查看即可

主服务器：

再授权一个用户，用于连接主从，给oneproxy使用

grant all on \*.\* to 'test'@'192.168.10.%' identified by '123.com';



oneproxy服务器

oneprox修改/usr/local/oneproxy/conf/proxy.conf文件

修改地址为主从两台的地址，为MySQL集群的IP地址



通过不同端口接入数据库（可写可不写）

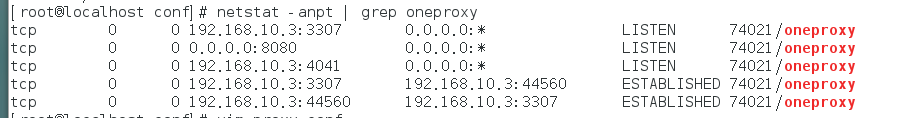
3307用于和后端MySQL集群进行通信，用于向oneproxy写入数据，数据通过策略向后端数据写入

4041是oneproxy的控制台端口，检测后端数据库状态，以及控制3307端口权限

8080是oneproxy可视化页面端口

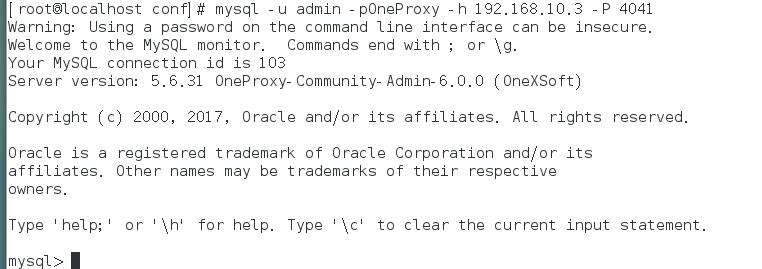


重启oneproxy，多重启几次（建议使用stop和start）

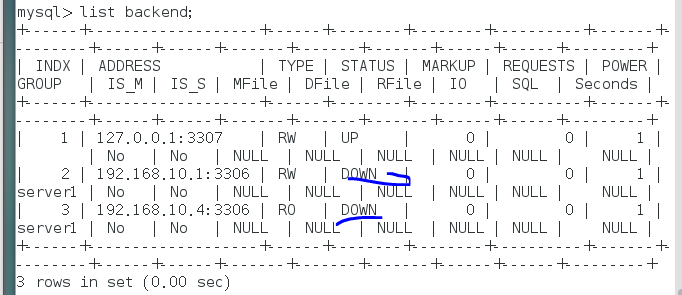


验证：

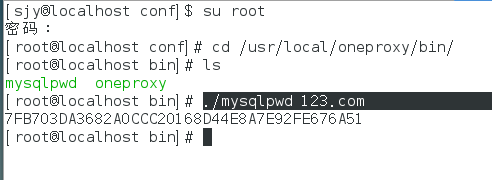
4041端口；用户和密码是自动生成的，控制台无法存放数据



list backend；查看集群状态



开第二个终端，cd到/usr/lcoal/oneproxy/bin，然后执行./mysqlpwd 123.com，生成hash值，复制该hash值

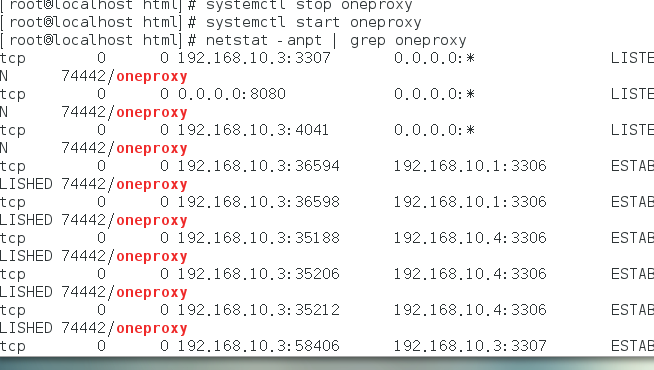


开第三终端，配置proxy.conf文件

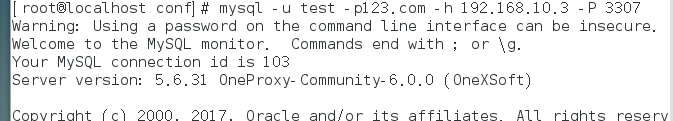
将复制的hash值添加到/后@前；@test：接入集群中的test库



重启oneproxy，netstat -anpt |grep oneproxy查看



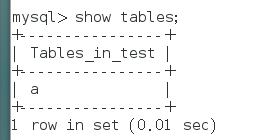
在第一终端登录3307端口



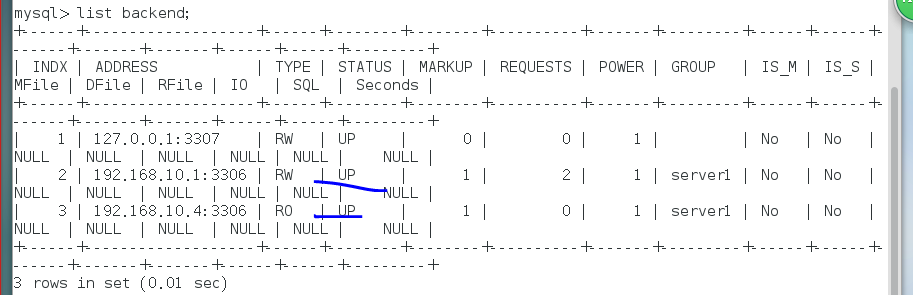


在主服务器上的test库中创建表

再在oneproxy服务器的3307端口上查看表



在oneproxy服务器的4041端口上查看listbackend；查看



在oneproxy的part.text文件中（分库分表策略）

全部删除

添加：

[

{

"table":"test", //分的表名（test库中的test表）

"pkey" :"id", //分的字段名（test表中的id字段）

"type" :"int", //分的字段的数据类型（id字段的int类）

"method":"hash", //根据hash的方式分表

"partitions": //切出的效果

[

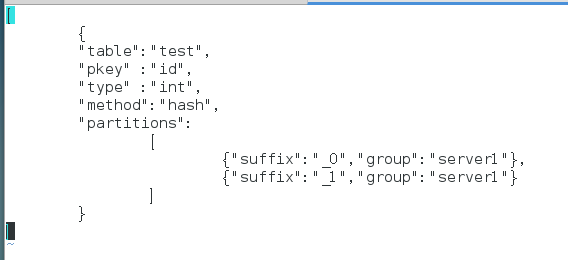
{"suffix":"\_0","group":"server1"},

{"suffix":"\_1","group":"server1"}

]

}

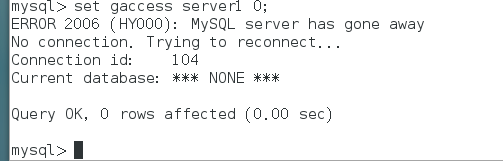
]



验证：在3307端口上创建表



在4041端口上



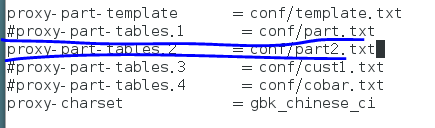
在3307端口上重新创表



分数据（二级分表）：

打开proxy.conf文件

开启第二个



vim part2.text

添加：

[

{

"table":"timeange",

"pkey" :"id",

"type" :"int",

"method":"crc32",

"partitions":

[

{"suffix":"\_0","group":"server1"},

{"suffix":"\_1","group":"server1"}

],

"subpkey":"tt",

"subtype":"timestamp",

"submethod":"range",

"subpartitions":

[

{"suffix":"\_2018","value":"2018-01-01 00:00:00"},

{"suffix":"\_2019","value":"2019-12-31 23:59:59"}

]

}

]

（先将timeange表按id以奇偶分为\_1,\_0两张表：奇数放在\_1表，偶数放在\_0表；

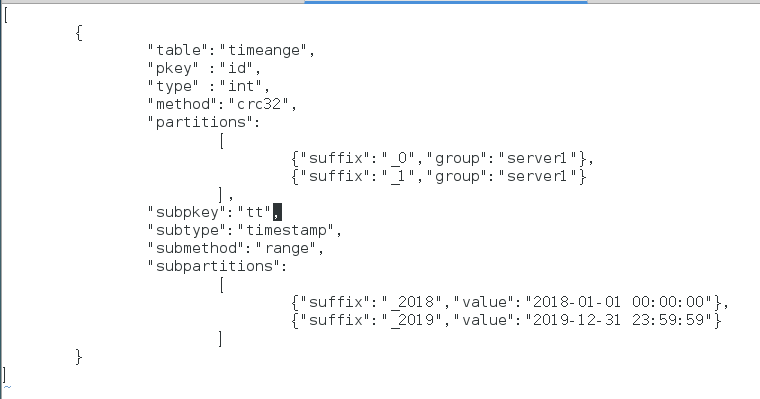
在根据二次切表，将分出的\_1,\_0表再次分为\_1\_2018,\_1\_2019,\_0\_2018,\_0\_2019四张表

按timestamp字段以时间分：2018-01-01 00:00:00之前且id为奇数的放在\_1\_2018表中；

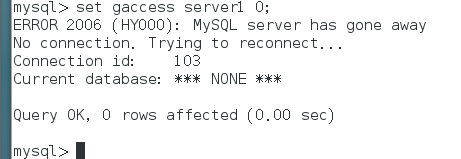
2018-01-01 00:00:00之前且id为偶数的放在\_0\_2018表中；

2019-12-31 23:59:59之前且id为奇数的放在\_1\_2019表中；

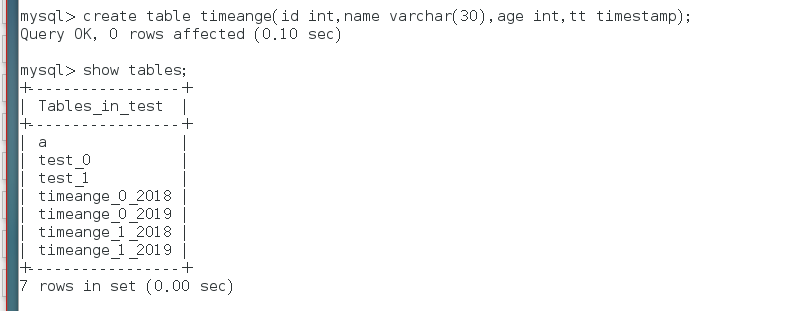
2019-12-31 23:59:59之前且id为偶数的放在\_0\_2019表中）



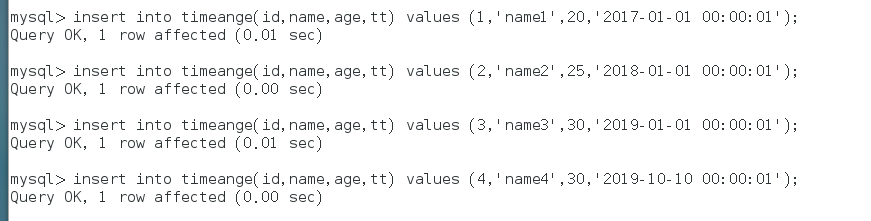
在4041端口上



在3307端口上（将表分为test\_0和test\_1；再将test\_0分为timeange\_0\_2018和timeange\_0\_2019，将test\_1分为timeange\_1\_2018，timeange\_1\_2019）



向timeange表中添加数据，会分散到四个表中



查看表即可

读写分离：

配置proxy.conf文件

添加：

proxy-group-policy=server1:2

设置读写分离策略：

1. 默认，不做设置，依靠lua脚本完成
2. read failover，先从主上读，如果主上不可用再去从上读取
3. read/weite split 先在从上读，从不可读，再去主上读
4. 对于xtraDB集群，选择固定的服务器进行写入数据，如果该服务器不可用在集群中挑选另一个作为固定节点
5. read/write split ，对于写，在主从随机选择一台，写在master写
6. 随机读写，读在主从任意一个随机选择，写在随机主上写

proxy-group-security=server1:0

设置集群安全级别

1. 不设置
2. 不允许oneproxy做ddl操作
3. 只允许执行带有where条件的语句
4. 只读操作

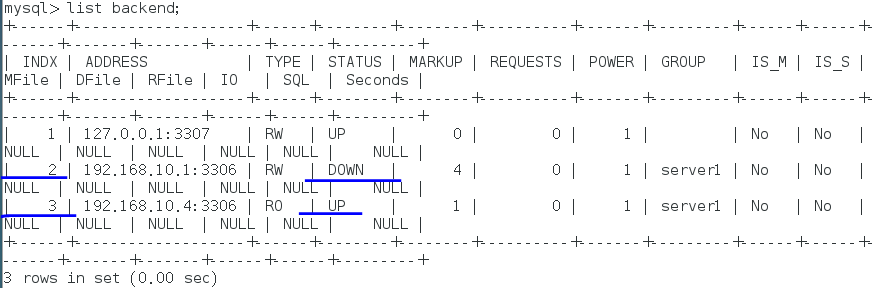


重启oneproxy

在主服务器上：断开主服务器

在oneproxy服务器上

在4041断开查看。主服务器已down



在3307断开查看表，可以查看，写入数据---无法写入；由此证明主down掉后，依旧可以在从服务器上读取数据，但无法写入。

