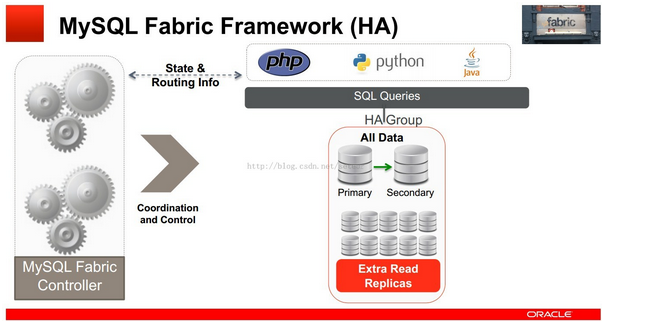
Mysql Fabric HA配置测试

MySQL Fabric是Oracle官方推出的原生mysql高可用性系统，它可以简化管理MySQL数据库群，它提供两大特性，一是通过故障检测和故障转移功能提供了高可用性，一是通过自动数据分片功能实现可扩展性。本文档只说明自动故障转移-高可用性的实现。



# 测试环境

|  |  |  |
| --- | --- | --- |
| 功能 | IP | 端口 |
| Master | 172.17.128.147 | 3306 |
| Slave | 172.17.128.148 | 3306 |
| Fabric state store | 172.17.128.146 | 3306 |
| Fabric 管理进程 | 172.17.128.146 | 32274 |

 系统环境要求如下：

|  |
| --- |
| MySQL Servers (version 5.6.10 or later)   * Backing store database server * Application database servers   Python 2.6 or 2. （ ·No support for 3.x yet）  python连接器1.2.1以上 |

# 安装连接器和MySQL Utilities

|  |
| --- |
| --安装连接器  tar -zxvf mysql-connector-python-2.1.3.tar.gz  cd mysql-connector-python-2.1.3  python setup.py install |
| --安装MySQL Utilities  tar -zxvf mysql-utilities-1.5.6.tar.gz  cd mysql-utilities-1.5.6  python setup.py install |
| #测试  mysqlfabric |

# 建立Backing Store帐号

Backing Store用于存储整个HA集群的服务器等相关配置，它需要一个MySQL实例来存储这些信息，这个实例的版本需要跟其它在HA中的MySQL实例版本保持一致，而且必须是5.6.10及更高的版本，我们在本例中选择172.17.128.146:3306实例来使用。Fabric管理进程和Backing Store实例可以是在两台不同的机器，测试中我们共用172.17.128.146。创建一个帐号来连接Backing Store的MySQL实例，这个帐号需要有对fabric数据库的管理员级权限，具体如下：

|  |
| --- |
| GRANT ALL ON fabric.\* TO fabric@'%' IDENTIFIED BY 'fabric@123';  FLUSH PRIVILEGES; |

# Fabric配置文件

Fabric配置文件默认位置是：/etc/mysql/fabric.cfg

修改其中的[storage]部分，具体如下：

|  |
| --- |
| [storage]  auth\_plugin = mysql\_native\_password  database = fabric  user = fabric    #连接state store实例fabric库的用户名  address = 172.17.128.146:3306       #指定state store的mysql实例地址和端口  connection\_delay = 1      #连续尝试连接间隔，单位秒  connection\_timeout = 6    #连接超时时间，单位秒  password = fabric@123     #用户密码  connection\_attempts = 6   #连接尝试连接的次数  修改其中的[servers]部分，具体如下：  [servers]  password = fabric@123     #连接HA组中实例的用户密码  user = fabric    #连接HA组中各实例的用户名  修改其中的[protocol.xmlrpc]部分，具体如下：  [servers]  user = admin     #管理账号  address = localhost:32274                   #fabric进程的服务器地址及端口号  password = abc123     #管理账号密码 |

# 填充Backing Store信息

我们通过执行以下命令来生成fabric数据库，具体如下：

|  |
| --- |
| [root@vdle-commerce63 mysql-utilities-1.5.6]# mysqlfabric manage setup  [INFO] 1460367552.524385 - MainThread - Initializing persister: user (fabric), server (172.17.128.146:3306), database (fabric).  Finishing initial setup  =======================  Password for admin user is not yet set.  Password for admin/xmlrpc:  Repeat Password:  Password set.  Password set.  期间会提示Fabric的管理员帐户admin没有设置密码，按提示设置密码。 |
| 连接state store实例可以看到新创建了一个fabfic数据库及相关存储配置信息的表。：show tables from fabric ;  +-------------------+  | Tables\_in\_fabric  |  +-------------------+  | checkpoints       | #存储程序执行信息，在crash后，能安全的恢复执行程序  | error\_log         | #服务器错误报告信息  | group\_replication | #定义复制，global groups和分片组，主要用于shard splitting,moving和global updates  | group\_view        |  | groups            | #包含管理组信息  | log               |  | permissions       | #包含权限信息，访问到不同fabric子系统，当前仅仅定义了core子系统  | proc\_view         |  | role\_permissions  | #表关联的角色和权限  | roles             | #包含用户角色的信息  | servers           | #包含fabric管理的所有服务器信息  | shard\_maps        | #包含名字和分片属性的映射  | shard\_ranges      | #分片索引和使用映射分片key到分片  | shard\_tables      | #所有分片的表  | shards            | #存储每个分片标识  | user\_roles        |  | users             | #标识用户有什么权限访问到不同子系统的功能 |

# 配置HA中主从MySQL节点帐号

我们需要建个管理员权限的帐号，该账号将用于fabric连接HA中的各实例，以及用于replication。注意帐号名也要跟fabric state store实例保持一致，本例中需要是fabric，具体如下：

GRANT ALL ON \*.\* TO fabric@'%' IDENTIFIED BY 'fabric@123';

FLUSH PRIVILEGES;

# 启动fabric

mysqlfabric manage start &

mysqlfabric manage stop

# 建立HA服务器组

#建立服务组

mysqlfabric group create mygroup\_1

#添加HA组的成员

mysqlfabric group add mygroup\_1 172.17.128.147:3306

mysqlfabric group add mygroup\_1 172.17.128.148:3306

# 查看HA服务组的成员信息

mysqlfabric group lookup\_servers mygroup\_1

#选举一个主库

mysqlfabric group promote mygroup\_1

# 查看HA服务组的成员信息

mysqlfabric group lookup\_servers mygroup\_1

#激活故障自动切换, Fabric并不会自动将Secondary角色切换成Master角色，所以我们需要将HA配置成可以自动切换角色

mysqlfabric group promote mygroup\_1

#从FAULTY状态不允许直接变更变SECONDARY状态，必须先变更成spare状态（避免写和读访问，同时保证当故障切换时不会被选取），然后再变更成secondary。

mysqlfabric server set\_status 69a05e95-07b8-11e6-8532-fa163e34eb2e spare

mysqlfabric server set\_status 69a05e95-07b8-11e6-8532-fa163e34eb2e secondary

#移除组成员

mysqlfabric group lookup\_servers mygroup\_1

mysqlfabric group demote mygroup\_1

mysqlfabric group remove mygroup\_1 07b5a6d1-06e8-11e6-bfe2-fa163e93578e

mysqlfabric group remove mygroup\_1 69a05e95-07b8-11e6-8532-fa163e34eb2e

mysqlfabric group remove mygroup\_1

# 常用命令

|  |
| --- |
| mysqlfabric group create mygroup          #创建HA组  mysqlfabric group destroy mygroup        #删除HA组  mysqlfabric group add mygroup 192.168.247.132:3306     #添加组成员  mysqlfabric group demote mygroup\_1  mysqlfabric group remove mygroup c505ce10-9bc0-11e4-bfd4-000c290e7abe   #移出组成员  mysqlfabric group lookup\_servers mygroup  #查看组成员  mysqlfabric group promote mygroup      #选举master  mysqlfabric group activate mygroup        #激活自动故障转移  mysqlfabric group deactivate mygroup           #禁用自动故障转移  mysqlfabric serverset\_status server\_uuid status #变更服务器状态  mysqlfabric help manage   #manage命令帮助  mysqlfabric help group     #group命令帮助  mysqlfabric help server      #server命令帮助 |

# 测试脚本

|  |
| --- |
| import sys  import mysql.connector  from mysql.connector import fabric    def main():     try:         conn = mysql.connector.connect(             fabric ={"host":"localhost","port":32274,"username":"admin",                       "password":"abc123"},             user = 'root',database = 'test',password = 'abc123',autocommit =True         )     except Exception,e:         print e         sys.exit(1)     else:         conn.set\_property(mode=fabric.MODE\_READWRITE,group="mygroup")         cur = conn.cursor()         cur.execute(             "CREATE TABLE IF NOT EXISTS t1 ("         " id INT,"         " first\_name varchar(40),"         " last\_name varchar(40)"         ")"         )     finally:         conn.close()    if \_\_name\_\_ == '\_\_main\_\_':     main()        2.插入数据测试脚本：  #test2.py  import sys  import mysql.connector  from mysql.connector import fabric    def add\_subscriber(conn,id,first\_name,last\_name):     conn.set\_property(group="mygroup\_1",mode=fabric.MODE\_READWRITE)     cur = conn.cursor()     cur.execute(         "INSERT INTO t1 VALUES (%s,%s,%s)",                   (id,first\_name,last\_name)      )    def main():     try:         conn = mysql.connector.connect(             fabric ={"host":"localhost","port":32274,"username":"admin",                       "password":"abc123"},             user="root",database="test",password="abc123",autocommit=True         )     except Exception,e:         print e         sys.exit(1)     else:         #conn.set\_property(group="mygroup",mode=fabric.MODE\_READWRITE)         add\_subscriber(conn, 1, "Tom", "Jerry")         add\_subscriber(conn, 2, "Jack", "Tom")         add\_subscriber(conn, 3, "Lily", "Lee")     finally:         conn.close()    if \_\_name\_\_ == '\_\_main\_\_':     main() |