MySQL/MariaDB数据同步服务DTS部署手册

准备工作

首先准备MySQL，Zookeeper和Kafka的环境。

MySQL

安装过程略（推荐使用see 默认的数据库）

MySQL需要开启binlog

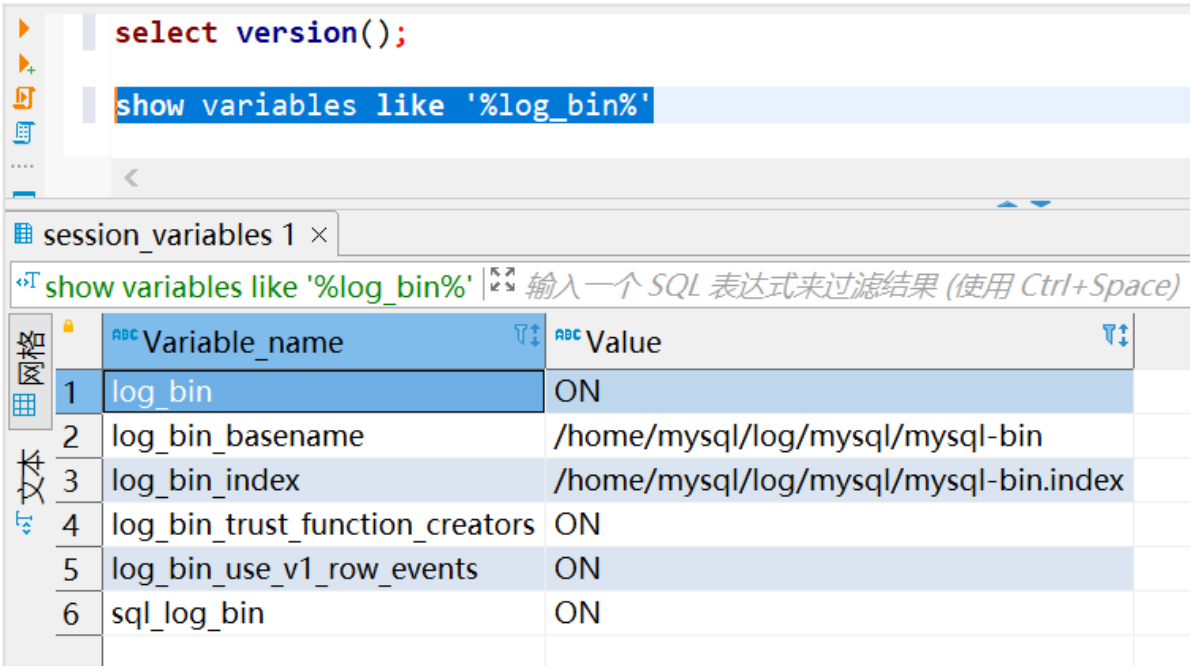
```
sudo vim /etc/my.cnf

[mysqld]
server-id=1
log-bin=mysql-bin
binlog_format=row

-- 改完配置文件以后，需要重启mysql
-- sudo systemctl restart mysqld

-- 然后通过查看变量，查看更改是否生效
-- show variables like '%log_bin%'
```

显示如下，说明MySQL准备就绪：



Zookeeper

下载地址：

https://dlcdn.apache.org/zookeeper/zookeeper-3.8.0/apache-zookeeper-3.8.0-bin.tar.gz

Kafka

下载地址：

https://downloads.apache.org/kafka/3.3.1/kafka_2.12-3.3.1.tgz

Debezium

下载Debezium mysql连接器：

https://repo1.maven.org/maven2/io/debezium/debezium-connector-mysql/1.9.7.Final/debezium-connector-mysql-1.9.7.Final-plugin.tar.gz

修改配置文件

Zookeeper

```
-- 解压
tar -zxvf apache-zookeeper-3.8.0-bin.tar.gz

-- 新建目录文件，存放zk数据文件，示例如下
mkdir zk-data

-- 复制配置文件
```

```
cd conf/
cp zoo_sample.cfg zoo.cfg

-- 编辑配置文件，指定zk数据目录，使用上面创建的目录“zk-data”
vim zoo.cfg

-- 这里是举例，请根据实际修改
dataDir=/data/lightdb/zm/apache-zookeeper-3.8.0-bin/zk-data
```

Kafka

```
-- 解压kafka
tar -zxvf kafka_2.12-3.3.1.tgz
-- 解压debezium插件
unzip debezium-debezium-connector-mysql-1.9.7.zip

-- 新建目录文件，用于存放kafka数据文件，示例如下
mkdir kafka-data

-- 再新建目录文件，用于存放kafka插件目录，示例如下
mkdir kafka-plugin
-- 然后把解压后的debezium文件夹，移动到新建的目录“kafka-plugin”中
mv debezium-debezium-connector-mysql-1.9.7 kafka-plugin/

-- 上面的操作完成之后，目录结构应该是这样
--kafka_2.12-3.3.1
    --config
    --bin
    --kafka-data
    --kafka-plugin
        --debezium-debezium-connector-mysql-1.9.7


-- 编辑kafka启动文件，指定kafka数据目录，使用上面创建的目录“kafka-data”
vim server.properties
-- 这里是举例，请根据实际修改
log.dirs=/data/lightdb/zm/kafka_2.12-3.3.1/kafka-data
-- 这里是举例，请根据实际修改成kafka部署所在IP和端口
listeners=PLAINTEXT://你的IP:9092


-- 编辑kafka连接器配置文件
vim connect-distributed.properties
-- 这里是举例，请根据实际修改成kafka部署所在IP和端口
bootstrap.servers=你的IP:9092
listeners=HTTP://你的IP:8083
-- 配置插件路径
plugin.path=/data/lightdb/zm/kafka_2.12-3.3.1/kafka-plugin
```

启动插件

Zookeeper

```
cd apache-zookeeper-3.8.0-bin/bin
./zkServer.sh start
```

Kafka

```
cd kafka_2.12-3.3.1/bin
./kafka-server-start.sh -daemon ../config/server.properties
```

Kafka连接器

```
cd kafka_2.12-3.3.1/bin
./connect-distributed.sh -daemon ../config/connect-distributed.properties
```

上面的步骤都执行好了之后，会存在3个进程，分别是ZK，Kafka和connectDistributed

Debezium请求配置

注意，下面http请求中的IP都是Kafka连接器所在的机子IP，也可以理解成Kafka所在机子的IP。

查看状态

可以直接在浏览器直接输入如下地址，查看debezium状态

```
http://IP:8083/
-- 正常则会返回如下类似信息
{"version":"3.3.1","commit":"e23c59d00e687ff5","kafka_cluster_id":"DL28VFkTSw-
_RgbMtutGXA"}

-- 也可以采用curl的方式查看
-- curl -H "Accept:application/json" IP:8083/
```

查看配置了哪些连接器

可以直接在浏览器直接输入如下地址，查看配置了哪些连接器

```
http://IP:8083/connectors
-- 没有配置则会返回空

-- 也可以采用curl的方式查看
curl -H "Accept:application/json" IP:8083/connectors/
```

注册MySQL连接器(最重要！！！)

可以使用postman，方便一些，带上下面的json参数来注册MySQL的连接器，需要设置请求头

```
POST请求
Accept=application/json
Content-Type=application/json


-- 消息体如下
{
    "name": "mariadb-connector",//名字可以自定义
    "config": {
        "connector.class": "io.debezium.connector.mysql.MySqlConnector",
        "tasks.max": "1",
        "database.hostname": "改成你的Mariadb数据库IP，示例10.20.30.40",
        "database.port": "3306",//改成实际数据库端口
        "database.user": "root",//改成实际数据库用户
        "database.password": "xx",//改成实际数据库密码
        "database.server.id": "184077",//数字可以自定义
        "database.server.name": "acm",//改成实际数据库服务名
        "database.include.list": "acm",//改成实际数据库服务名
        "database.history.kafka.bootstrap.servers": "改成你的kafka信息，示例
10.20.30.40:9092",
        "database.history.kafka.topic": "schema-changes.acm",//后面的.acm改成实际数
据库服务名
        "include.schema.changes": "true",
        "decimal.handling.mode": "string",
        "include.schema.comments": "false"
    }
}


-- 如果没有postman，则可以使用curl来发送post请求，示例如下：
curl -i -X POST -H "Accept:application/json" -H "Content-Type:application/json"
你的IP:8083/connectors/ -d '{
    "name": "mariadb-connector",
    "config": {
        "connector.class": "io.debezium.connector.mysql.MySqlConnector",
        "tasks.max": "1",
        "database.hostname": "127.0.0.1",
        "database.port": "3306",
        "database.user": "root",
        "database.password": "xxx",
        "database.server.id": "184077",
        "database.server.name": "acm",
```

```
        "database.include.list": "acm",
        "database.history.kafka.bootstrap.servers": "127.0.0.1:9092",
        "database.history.kafka.topic": "schema-changes.acm",
        "include.schema.changes": "true",
        "decimal.handling.mode": "string",
        "include.schema.comments": "false"
    }
}'
```

连接器状态检查

比如上面注册的连接器名字为mariadb-connector，直接在浏览器输入地址，就可以查看连接器信息

```
http://IP:8083/connectors/mariadb-connector/status

-- 如果要在服务器上操作
curl -i IP:8083/connectors/mariadb-connector/status
```

查看连接器配置

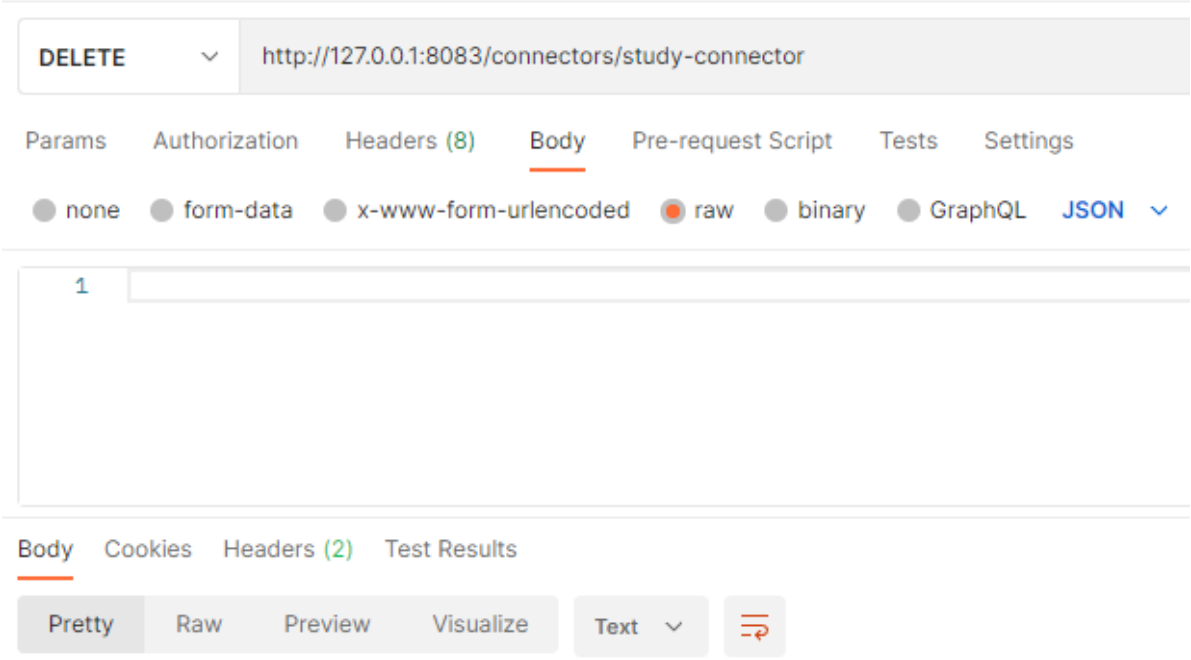
还是以上面注册的mariadb-connector举例，在浏览器输入如下地址，查看注册的配置信息

```
http://IP:8083/connectors/mariadb-connector/config

-- 如果要在服务器上操作
curl -i IP:8083/connectors/mariadb-connector/config
```

删除连接器

可以使用postman发起delete请求，后面跟上注册器的名字，如下图所示




```
-- 如果要在服务器上操作
curl -v -X DELETE IP:8083/connectors/mariadb-connector
```

启动数据同步服务

注意：需要JDK11下运行

启动示例如下：

```
nohup java -jar ltdts_mysql.jar --server.port=8888 --zk=127.0.0.1:2181 --
kafka=127.0.0.1:9092 --lh=127.0.0.1:5432/lt_test --lu=lightdb --lp=lightdb123 --
bl=tb_lock,tb_dssp_see_platform_node > log.file  2>&1 &
```

命令解释如下：

命令	解释	默认值
--server.port	同步服务端口，比如8080	8080
--zk	zookeeper连接信息，IP和端口，比如127.0.0.1:2181	
--kafka	kafka连接信息，IP和端口，比如127.0.0.1:9092	
--db	同步的数据库，默认是see的数据库	acm
--lh	LightDB数据库信息，比如127.0.0.1:5432/lt_test	

命令	解释	默认值
--lu	LightDB用户	
--lp	LightDB密码	
--bl	黑名单，多个以英文逗号分隔，比如table1,table2	