Debezium Connector
配置说明文档

文件修订记录

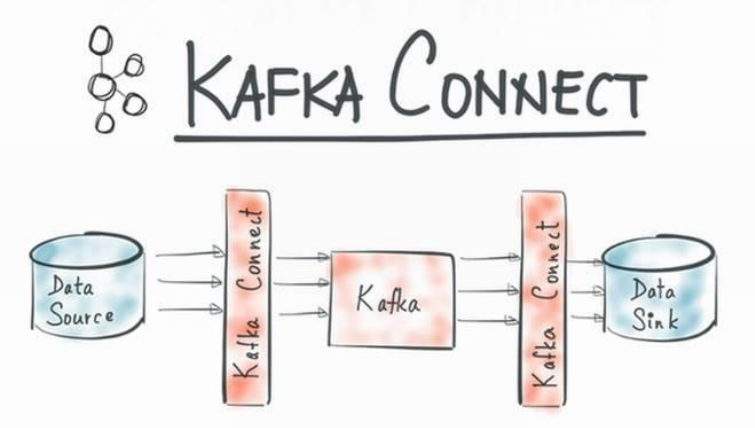
文档名称	云平台 - Debezium Connector 配置说明文档		
文档编号			
所属部门	BUPT-ICC 云平台开发小组		
密级程度	☐绝密 ☑对内 ☐可公开		
修订日期	修订描述	修订人	版本号
2017/7/11	创建文件初稿	韩吉雯	v1.0.0

一、 Kafka Connect

1 简介

Kafka Connect 是一个为 Kafka 和外部系统提供可靠的、可扩展的连接的框架。比如可连接的外部系统有数据库、键-值存储、搜索索引、文件系统等等。

2 整体框架



3 整体框架说明

围绕 Kafka，Kafka Connect 旨在建造一个可伸缩、可靠的数据流通道，通过 Kafka Connect 可以实现一个数据源和目标数据源之间的进行大量数据进出的数据管道。

4 Debezium Connector

Debezium 是基于 Kafka Connect 建立的开源连接器。选择 MySQL 作为数据源，通过记录数据库历史操作，监测数据库中增加、删除、修改的这几类操作。通过消费 Kafka 的特定主题，就可以实时监测数据库的变化。并且即使连接器关闭或重启，也不会丢失任一条消息。

二、 配置说明

1. 前期准备工作

1）开启 zookeeper，默认端口为 2181

开启 zookeeper （此处使用 confluent 内置 zookeeper）

```
    ./bin/zookeeper-server-
start ./etc/kafka/zookeeper.properties >> ../zookeeper/zookeeper.log &
```

（设置后台运行，输出日志文件位于 zookeeper/下）

2）开启 Kafka broker，默认端口为 9092

开启 Kafka broker （此处使用 confluent 内置 kafka）

```
    ./bin/kafka-server-start ./etc/kafka/server.properties
```

3）配置 MySQL 开启 binlog

修改 MySQL 配置文件

```
server-id=2331
log_bin=mysql-bin
binlog_format=row
```

server-id 与 connector 中配置一致即可

重启 MySQL

检查是否开启 binlog，值为 OFF 代表关闭，值为 ON 代表开启

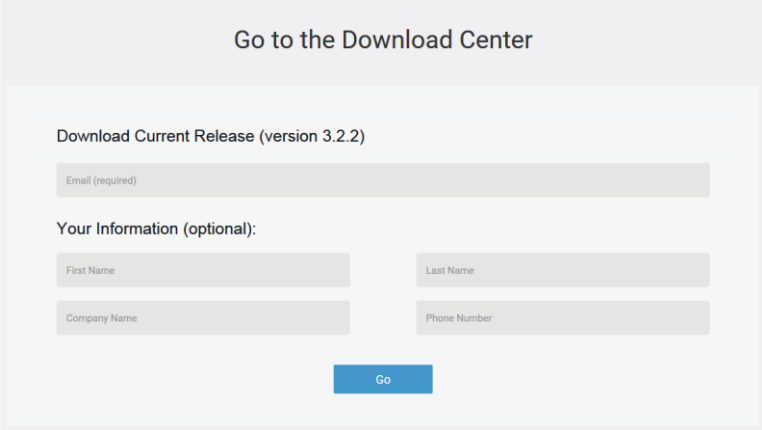
```
mysql> show variables like 'log_bin';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_bin       | ON    |
+---------------+-------+
1 row in set (0.01 sec)
```

4） 开启 Kafka Connect

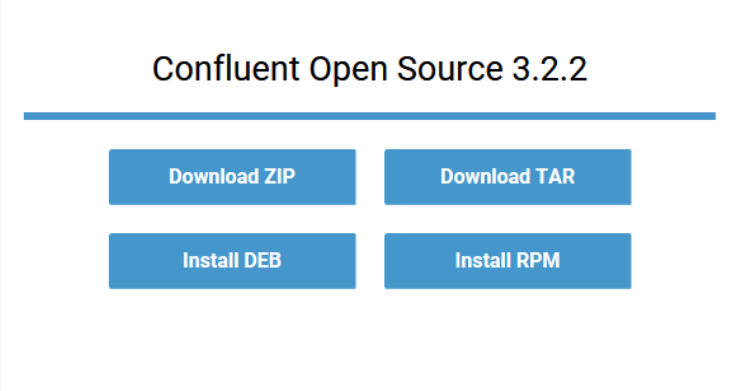
（代码包中已包含 confluent 文件，可跳过下载配置过程，直接开启 Kafka Connect）

下载 Confluent

```
https://www.confluent.io/download/
```



输入邮箱，点击 Go



选择 Confluent Open Source 3.2.2,下载压缩包

解压

```
tar -xzvf confluent-oss-3.2.1-2.11.tar.gz
```

解压后目录下文件

```
bin/          可执行脚本文件
etc/          配置文件
share/        jar 包等额外配置文件
src/          其他资源压缩包
logs/         日志文件（运行后产生）
```

修改 Kafka Connect 配置文件

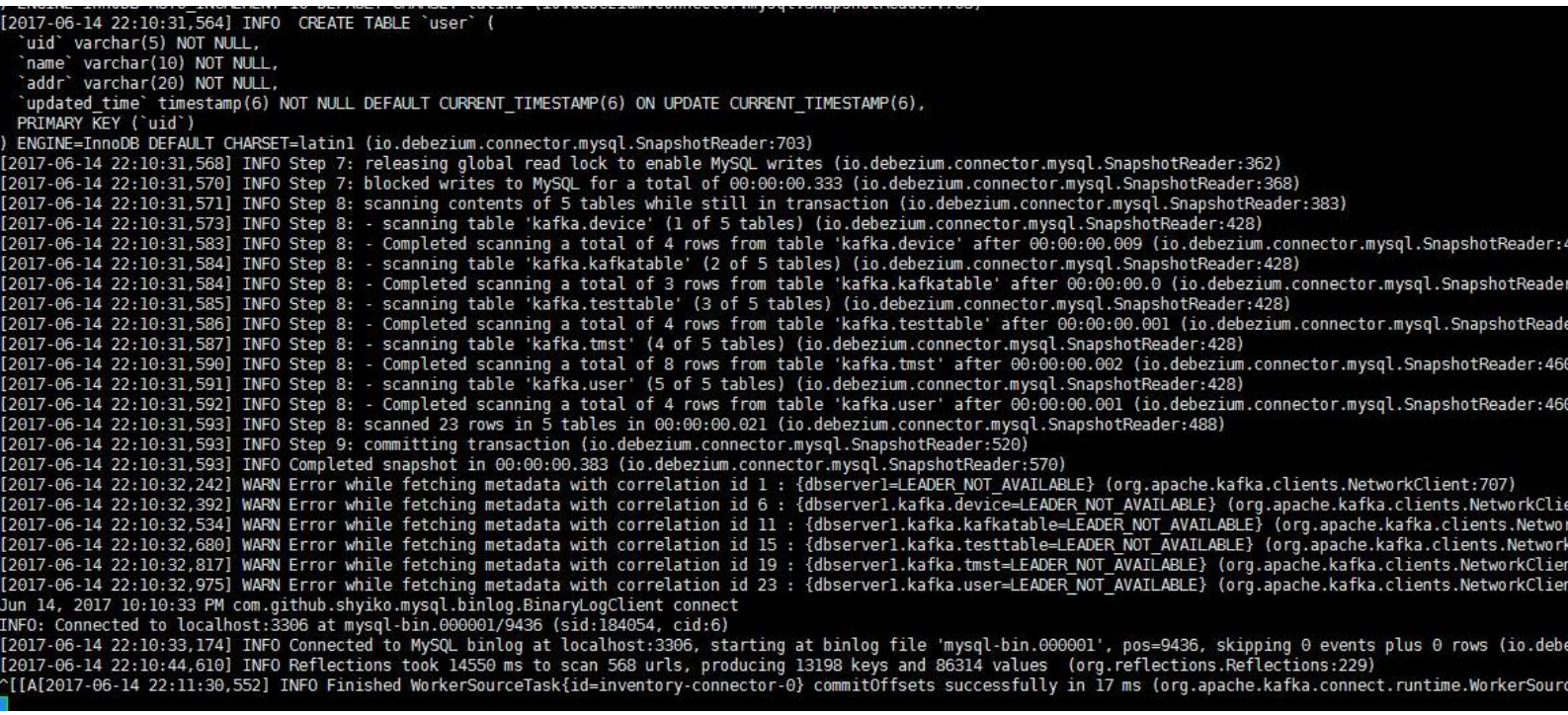
etc/schema-registry/connect-avro-distributed.properties

```
key.converter = org.apache.kafka.connect.json.JsonConverter

value.converter =  org.apache.kafka.connect.json.JsonConverter
```

开启 Kafka Connect （distributed 模式）

```
./bin/connect-distributed  ./etc/schema-registry/connect-avro-distributed.properties
```



检查 Kafka Connect 状态（默认本地开启，端口 8083）

```
curl -H "Accept:application/json" localhost:8083/
```

若未开启，则连接失败

```
curl: (7) Failed to connect to localhost port 8083: Connection refused
```

若成功开启，则可查看 Kafka Connect 版本信息

```
{"version":"0.10.2.1-cp1","commit":"078e7dc02a100018"}
```

列出当前所有注册的 connector

```
curl -H "Accept:application/json" localhost:8083/connectors/
```

或者

```
curl -X GET localhost:8083/connectors
```

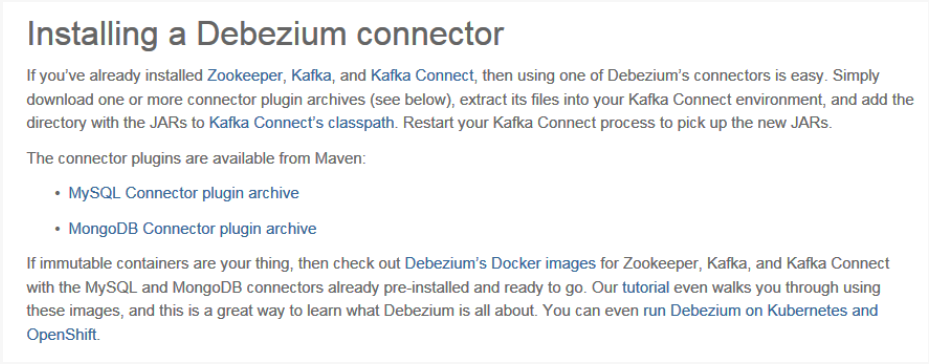
结果显示所有 connector 的名称列表，若无，则为"[]"空列表

5） 注册 Debezium Connector

（代码包中 confluent 文件夹下已包含 debezium 插件，可跳过下载拷贝过程，直接开启 Kafka Connect，并注册连接器）

下载 Debezium Connector plugin archives

http://debezium.io/docs/install/

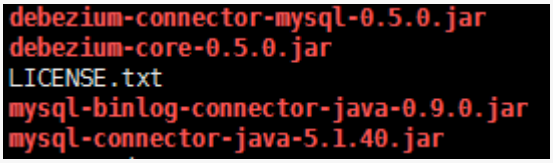


选择 MySQL Connector plugin archive 下载

解压

```
tar -xzvf debezium-connector-mysql-0.5.0-plugin.tar.gz
```

将解压后所有 jar 包：



复制至 confluent 目录，share/java/ kafka-connect-jdbc 位置下

重启 Kafka Connect

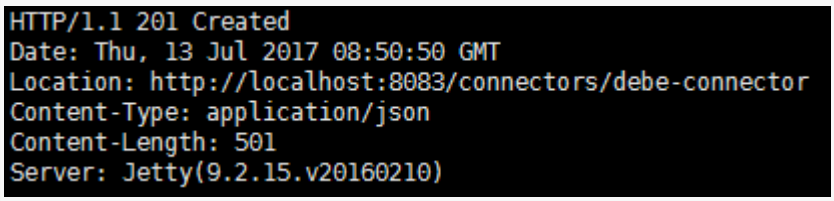
否则在注册 Debezium Connector 时，将无法找到相关类

（也可先进行上述拷贝操作，再直接开启 Kafka Connect）

注册新的 Debezium Connector，其中 JSON 格式配置按下方配置信息设置

```
curl -i -X POST -H "Accept:application/json" -H "Content-
Type:application/json" localhost:8083/connectors/ -d '{ "name":
"debe-connector", "config": { "connector.class":
```

```
    "io.debezium.connector.mysql.MySqlConnector", "tasks.max": "1",
    "database.hostname": "localhost", "database.port": "3306",
    "database.user": "root", "database.password": "xxx",
    "database.server.id": "2331", "database.server.name": "dbserver",
    "database.whitelist": "kafka",
    "database.history.kafka.bootstrap.servers": "localhost:9092",
    "database.history.kafka.topic": "dbhistory.debetopic" } }'
```



注册成功！返回 connector 信息

具体 connector 配置信息

```
{
  "name": "debe-connector", //connector 名称
  "config": {
     "connector.class": "io.debezium.connector.mysql.MySqlConnector",
     "tasks.max": "1",
     "database.hostname": "localhost",
     "database.port": "3306", // 数据库地址及端口
     "database.user": "root",
     "database.password": "xxx", // 数据库用户名及密码
     "database.server.id": "2331", // 数据库 id，与 mysql 配置 id 一致
     "database.server.name": "dbserver", //设定 topic 前缀
     "database.whitelist": "kafka", //mysql 中监听的 database 白名单
     "database.history.kafka.bootstrap.servers": "localhost:9092", // 入口
broker 地址
     "database.history.kafka.topic": "dbhistory.debetopic",
```

```
  }

}
```

其他详细配置见 http://debezium.io/docs/connectors/mysql/#configuration

涉及 topic

```
1. dbhistory.debetopic    由 database.history.kafka.topic 配置
2. dbserver               由 database.server.name 配置，表示 schema change topic
3. dbserver.kafka.xxx     由 database.server.name 作为前缀，kafka 为 database
   name，后缀加上 table name：xxx，表示数据库中表有增、删、改变化的 topic
```

查看 connector 信息或状态

```
curl -X GET localhost:8083/connectors/debe-connector

curl -X GET localhost:8083/connectors/debe-connector/status
```

{"name":"debe-connector","connector":{"state":"RUNNING","worker_id":"127.0.1
.1:8083"},"tasks":[{"state":"RUNNING","id":0,"worker_id":"127.0.1.1:8083"}]}

6） 运行同步代码

代码包 debezium-mysql-connector 文件目录

```
bin/                    可执行脚本文件
config/                 配置文件
lib/                    核心代码
models/                 数据库封装操作
node_modules/           node 模块
confluent-3.2.1/        confluent 文件，包含 kafka connect
zookeeper/              zookeeper log 文件（若单独开启 zookeeper，则
可忽略）
package.json            npm 文件
pm2.json                pm2 文件
```

安装 node 模块

```
npm install
根据 package.json 自动安装所有依赖
```

根据所需监测的数据库表配置 config/config.json 文件

```
{
  "topics": {
    "serverName": "dbserver",
    "databaseName": "kafka",
    "tableWhiteList": ["Users", "Devices", "Groups",
      "User2Devices", "Group2Devices",
      "Developers", "Products", "OTAs",
      "Developer2Pros", "Product2OTAs"]
  },
    "options": {
      "host": "localhost:2181",
      "ssl": false,
      "groupId": "SyncGroup",
      "id": "consumer1",
      "protocol": ["roundrobin"],
      "fromOffset": "latest",
      "outOfRangeOffset": "earliest",
      "migrateHLC": false,
      "migrateRolling": true
    }
}
```

1. 配置数据库名称 databaseName
2. 配置监测的表格 tableWhiteList，列表形式

运行同步程序

```
pm2 start pm2.json
```

```
[PM2][WARN] You are starting 1 processes in fork_mode without load balancing. To enable it remove -x o
[PM2] Applying action restartProcessId on app [debezium-mysql-connector](ids: 2)
[PM2] [debezium-mysql-connector](2) ✓
```

App name	id	mode	pid	status	restart	uptime	cpu	mem	watchin
debezium-mysql-connector	2	fork	30119	online	0	0s	0%	11.7 MB	disable

查看 log 文件

```
tail -f /tmp/debezium-mysql-connector.log
```

```
connected.

coming message...
consumer1 from Topic=dbserver.kafka.Users Partition=0 Offset=12
Operation:  updateUsers.
connected.
```

（日志文件地址可在 pm2.json 中配置）

三、参考网址

http://debezium.io/

http://docs.confluent.io/3.2.1/

http://docs.confluent.io/current/connect/quickstart.html