基于Canal与Flink实现数据实时增量同步(一)

canal是阿里巴巴旗下的一款开源项目，纯Java开发。基于数据库增量日志解析，提供增量数据订阅&消费，目前主要支持了MySQL（也支持mariaDB）。

准备

常见的binlog命令

```
# 是否启用binlog日志
show variables like 'log_bin';
# 查看binlog类型
show global variables like 'binlog_format';
# 查看详细的日志配置信息
show global variables like '%log%';
# mysql数据存储目录
show variables like '%dir%';
# 查看binlog的目录
show global variables like "%log_bin%";
# 查看当前服务器使用的biglog文件及大小
show binary logs;
# 查看最新一个binlog日志文件名称和Position
show master status;
```

配置MySQL的binlog

对于自建 MySQL , 需要先开启 Binlog 写入功能，配置 binlog-format 为 ROW 模式，my.cnf 中配置如下

```
[mysqld]
log-bin=mysql-bin # 开启 binlog
binlog-format=ROW # 选择 ROW 模式
server_id=1 # 配置 MySQL replaction 需要定义，不要和 canal 的 slaveId 重复
```

授权

授权 canal 链接 MySQL 账号具有作为 MySQL slave 的权限, 如果已有账户可直接 grant

```
CREATE USER canal IDENTIFIED BY 'canal';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%
```

```
';
-- GRANT ALL PRIVILEGES ON *.* TO 'canal'@'%' ;
FLUSH PRIVILEGES;
```

部署canal

安装canal

- 下载：点此下载
- 解压缩

```
[kms@kms-1 softwares]$ tar -xzvf canal.deployer-1.1.4.tar.gz  -C /opt/modules/canal/
```

- 目录结构

```
drwxr-xr-x 2 root root 4096 Mar  5 14:19 bin
drwxr-xr-x 5 root root 4096 Mar  5 13:54 conf
drwxr-xr-x 2 root root 4096 Mar  5 13:04 lib
drwxrwxrwx 4 root root 4096 Mar  5 14:19 logs
```

配置修改

- 修改conf/example/instance.properties，修改内容如下：

```
## mysql serverId
canal.instance.mysql.slaveId = 1234
#position info, 需要改成自己的数据库信息
canal.instance.master.address = kms-1.apache.com:3306
#username/password, 需要改成自己的数据库信息
canal.instance.dbUsername = canal
canal.instance.dbPassword = canal
# mq config, kafka topic名称
canal.mq.topic=test
```

- 修改conf/canal.properties，修改内容如下：

```
# 配置zookeeper地址
canal.zkServers =kms-2:2181,kms-3:2181,kms-4:2181
# 可选项: tcp(默认), kafka, RocketMQ,
canal.serverMode = kafka
```

```
# 配置kafka地址
canal.mq.servers = kms-2:9092,kms-3:9092,kms-4:9092
```

启动canal

```
sh bin/startup.sh
```

关闭canal

```
sh bin/stop.sh

```

部署Canal Admin(可选)

canal-admin设计上是为canal提供整体配置管理、节点运维等面向运维的功能，提供相对友好的WebUI操作界面，方便更多用户快速和安全的操作。

要求

canal-admin的限定依赖：

- MySQL，用于存储配置和节点等相关数据
- canal版本，要求>=1.1.4 (需要依赖canal-server提供面向admin的动态运维管理接口)

安装canal-admin

- 下载

  点此下载

- 解压缩

```
[kms@kms-1 softwares]$ tar -xzvf canal.admin-1.1.4.tar.gz  -C /opt/modul
es/canal-admin/
```

- 目录结构

```
drwxrwxr-x 2 kms kms 4096 Mar  6 11:25 bin
drwxrwxr-x 3 kms kms 4096 Mar  6 11:25 conf
drwxrwxr-x 2 kms kms 4096 Mar  6 11:25 lib
```

```
drwxrwxr-x 2 kms kms 4096 Sep  2  2019 logs

```

- 配置修改

```
vi conf/application.yml

```

```
server:
  port: 8089
spring:
  jackson:
    date-format: yyyy-MM-dd HH:mm:ss
    time-zone: GMT+8

spring.datasource:
  address: kms-1:3306
  database: canal_manager
  username: canal
  password: canal
  driver-class-name: com.mysql.jdbc.Driver
  url: jdbc:mysql://${spring.datasource.address}/${spring.datasource.dat
abase}?useUnicode=true&characterEncoding=UTF-8&useSSL=false
  hikari:
    maximum-pool-size: 30
    minimum-idle: 1

canal:
  adminUser: admin
  adminPasswd: admin
```

- 初始化原数据库

```
mysql -uroot -p
# 导入初始化SQL
#注：(1)初始化SQL脚本里会默认创建canal_manager的数据库，建议使用root等有超级权限的
账号进行初始化
#    (2)canal_manager.sql默认会在conf目录下
> mysql> source /opt/modules/canal-admin/conf/canal_manager.sql
```

- 启动canal-admin

```
sh bin/startup.sh

```

- 访问

可以通过 http://kms-1:8089/ 访问，默认密码：admin/123456

- canal-server端配置

使用canal_local.properties的配置覆盖canal.properties,将下面配置内容配置在canal_local.properties文件里面，就可以了。

```
# register ip
canal.register.ip =
# canal admin config
canal.admin.manager = 127.0.0.1:8089
canal.admin.port = 11110
canal.admin.user = admin
canal.admin.passwd = 4ACFE3202A5FF5CF467898FC58AAB1D615029441
# admin auto register
canal.admin.register.auto = true
canal.admin.register.cluster =

```

- 启动canal-serve

```
sh bin/startup.sh  local

```

注意：先启canal-server,然后再启动canal-admin，之后登陆canal-admin就可以添加serve和instance了。

启动kafka控制台消费者测试

```
bin/kafka-console-consumer.sh --bootstrap-server kms-2:9092,kms-3:9092,k
ms-4:9092  --topic test --from-beginning

```

此时MySQL数据表若有变化，会将row类型的log写进Kakfa，具体格式为JSON：

- insert操作

```
{
    "data":[
        {
            "id":"338",
            "city":"成都",
            "province":"四川省"
        }
    ],
    "database":"qfbap_ods",
    "es":1583394964000,
    "id":2,
    "isDdl":false,
    "mysqlType":{
        "id":"int(11)",
        "city":"varchar(256)",
        "province":"varchar(256)"
    },
    "old":null,
    "pkNames":[
        "id"
    ],
    "sql":"",
    "sqlType":{
        "id":4,
        "city":12,
        "province":12
    },
    "table":"code_city",
    "ts":1583394964361,
    "type":"INSERT"
}
```


- update操作


```
{
    "data":[
        {
            "id":"338",
            "city":"绵阳市",
            "province":"四川省"
        }
    ],
    "database":"qfbap_ods",
    "es":1583395177000,
    "id":3,
```

```
    "isDdl":false,
    "mysqlType":{
        "id":"int(11)",
        "city":"varchar(256)",
        "province":"varchar(256)"
    },
    "old":[
        {
            "city":"成都"
        }
    ],
    "pkNames":[
        "id"
    ],
    "sql":"",
    "sqlType":{
        "id":4,
        "city":12,
        "province":12
    },
    "table":"code_city",
    "ts":1583395177408,
    "type":"UPDATE"
}
```


- delete操作


```
{
    "data":[
        {
            "id":"338",
            "city":"绵阳市",
            "province":"四川省"
        }
    ],
    "database":"qfbap_ods",
    "es":1583395333000,
    "id":4,
    "isDdl":false,
    "mysqlType":{
        "id":"int(11)",
        "city":"varchar(256)",
        "province":"varchar(256)"
    },
    "old":null,
    "pkNames":[
```

```
            "id"
        ],
        "sql":"",
        "sqlType":{
            "id":4,
            "city":12,
            "province":12
        },
        "table":"code_city",
        "ts":1583395333208,
        "type":"DELETE"
    }

```

JSON日志格式解释

- data：最新的数据，为JSON数组，如果是插入则表示最新插入的数据，如果是更新，则表示更新后的最新数据，如果是删除，则表示被删除的数据
- database：数据库名称
- es：事件时间，13位的时间戳
- id：事件操作的序列号，1,2,3…
- isDdl：是否是DDL操作
- mysqlType：字段类型
- old：旧数据
- pkNames：主键名称
- sql：SQL语句
- sqlType：是经过canal转换处理的，比如unsigned int会被转化为Long，unsigned long会被转换为BigDecimal
- table：表名
- ts：日志时间
- type：操作类型，比如DELETE，UPDATE，INSERT

小结

本文首先介绍了MySQL binlog日志的配置以及Canal的搭建，然后描述了通过canal数据传输到Kafka的配置，最后对canal解析之后的JSON数据进行了详细解释。本文是基于Canal与Flink实现数据实时增量同步的第一篇，在下一篇介绍如何使用Flink实现实时增量数据同步。