搭建maxwell

注意这里需要搭建mysql等

Mysql设置

2.1.3 MySQL 环境准备

（1）修改 mysql 的配置文件，开启 MySQL Binlog 设置

|  |
| --- |
| sudo vim /etc/my.cnf  在[mysqld]模块下添加一下内容  [mysqld]  server\_id=1  log-bin=mysql-bin  binlog\_format=row  #binlog-do-db=test\_maxwell |

并重启 Mysql 服务

|  |
| --- |
| sudo systemctl restart mysqld |

登录 mysql 并查看是否修改完成

|  |
| --- |
| mysql -uroot -p123456  show variables like '%binlog%'; |

查看下列属性

|  |
| --- |
| binlog\_format | ROW |

（2）进入/var/lib/mysql 目录，查看 MySQL 生成的 binlog 文件

|  |
| --- |
| cd /var/lib/mysql  sudo ls -l |

|  |
| --- |
| 总用量 188500  -rw-r-----. 1 mysql mysql 154 11 月 17 16:30 mysql-  bin.000001  -rw-r-----. 1 mysql mysql 19 11 月 17 16:30 mysql-  bin.index |

注：MySQL 生成的 binlog 文件初始大小一定是 154 字节，然后前缀是 log-bin 参数配置的，后缀是默从.000001，然后依次递增。除了 binlog 文件文件以外，MySQL 还会额外生产一个.index 索引文件用来记录当前使用的 binlog 文件。

2.1.4 初始化 Maxwell 元数据库

（1）在 MySQL 中建立一个 maxwell 库用于存储 Maxwell 的元数据

|  |
| --- |
| mysql -uroot -p123456  CREATE DATABASE maxwell; |

（2）设置 mysql 用户密码安全级别

|  |
| --- |
| set global validate\_password\_length=4;  set global validate\_password\_policy=0; |

（3）分配一个账号可以操作该数据库

|  |
| --- |
| GRANT ALL ON maxwell.\* TO 'maxwell'@'%' IDENTIFIED BY '123456'; |

（4）分配这个账号可以监控其他数据库的权限

|  |
| --- |
| GRANT SELECT ,REPLICATION SLAVE , REPLICATION CLIENT ON  \*.\* TO maxwell@'%'; |

（5）刷新 mysql 表权限

|  |
| --- |
| flush privileges; |

1. 解压maxwell-1.29.2.tar.gz

|  |
| --- |
| tar -zxvf maxwell-1.29.2.tar.gz -C /opt/model |

1. 跳转到flume下拷贝config.properties.example 名字为config.properties

|  |
| --- |
| cd /opt/maxwell-1.29.0  拷贝config.properties.example 名字为config.properties  cp config.properties.example config.properties |

1. 修改 config.properties

|  |
| --- |
| vi config.properties  配置文件内容  修改一下内容即可  kafka.bootstrap.servers=localhost:9092  # mysql login info  host=localhost  user=maxwell  password=123456 |

1. 启动maxwell并打印到终端上

|  |
| --- |
| bin/maxwell --user='maxwell' -password='123456' --host='slave1' --producer=stdout  解释如下：  --user：指定连接 MySQL 数据库使用的用户名。  -password：指定连接 MySQL 数据库使用的密码  --host：指定连接 MySQL 数据库所在的主机名或 IP 地址。  --producer：指定 Maxwell 的输出目标，这里是 stdout（标准输出） |

1. 启动并打印到Kafka中

|  |
| --- |
| bin/maxwell --user='maxwell' -password='123456' --host='slave1' --producer='kafka' --kafka.bootstrap.servers=slave1:9092 --kafka\_topic=ods\_mall\_data  解释  --user：指定连接 MySQL 数据库使用的用户名。  -password：指定连接 MySQL 数据库使用的密码。  --host：指定连接 MySQL 数据库所在的主机名或 IP 地址。  --producer：指定 Maxwell 的输出目标，这里是 Kafka。  --kafka.bootstrap.servers：指定连接 Kafka 集群的地址和端口。  --kafka\_topic：指定发送消息到 Kafka 的 topic 名称。 |

1. 打开对应的kafka

|  |
| --- |
| kafka-console-consumer.sh --bootstrap-server slave1:9092 --topic ods\_mall\_data  解释：  producer的值来控制输出到什stdout：将变更数据以 JSON 格式输出到标准输出（stdout）。  file：将变更数据以 JSON 格式输出到文件中，需要通过 --output\_file 参数指定输出的文件路径。  kafka：将变更数据发送到 Kafka 集群中，需要指定 Kafka 集群地址和端口以及要发送到的 Kafka topic 名称等参数。  pubsub：将变更数据发送到 Google Cloud Pub/Sub 中，需要指定 Pub/Sub 项目 ID、订阅 ID 和认证信息等参数。 |

注意：然后打开数据库输入东西就可以在以上命令的界面（注：他只能读取增删改）

1. 指定表更新

|  |
| --- |
| bin/maxwell --user='maxwell' -password='123456' --host='slave1' --producer='kafka' --filter 'exclude' \*.\*, include:'表名' --kafka.bootstrap.servers=slave1:9092 --kafka\_topic=ods\_mall\_data  解释如下：  bin/maxwell 是 Maxwell 的启动命令。  --user='maxwell' -password='123456' --host='slave1' 指定了 Maxwell 连接 MySQL 数据库的用户名、密码和主机名。  --producer='kafka' 表示使用 Kafka 作为 Maxwell 的数据生产者。  --filter 'exclude' \*.\*, include:'表名' 指定了同步规则，exclude 表示要排除的表，\*.\* 表示排除所有表，include:'表名' 表示要同步的表，可以指定多个表名，用逗号隔开。  --kafka.bootstrap.servers=slave1:9092 指定了 Kafka 的 brokers 地址和端口。  --kafka\_topic=ods\_mall\_data 指定了将数据发送到的 Kafka topic 名称 |