Trino导入使用的平台为

<http://gzcloud21:9090/main>

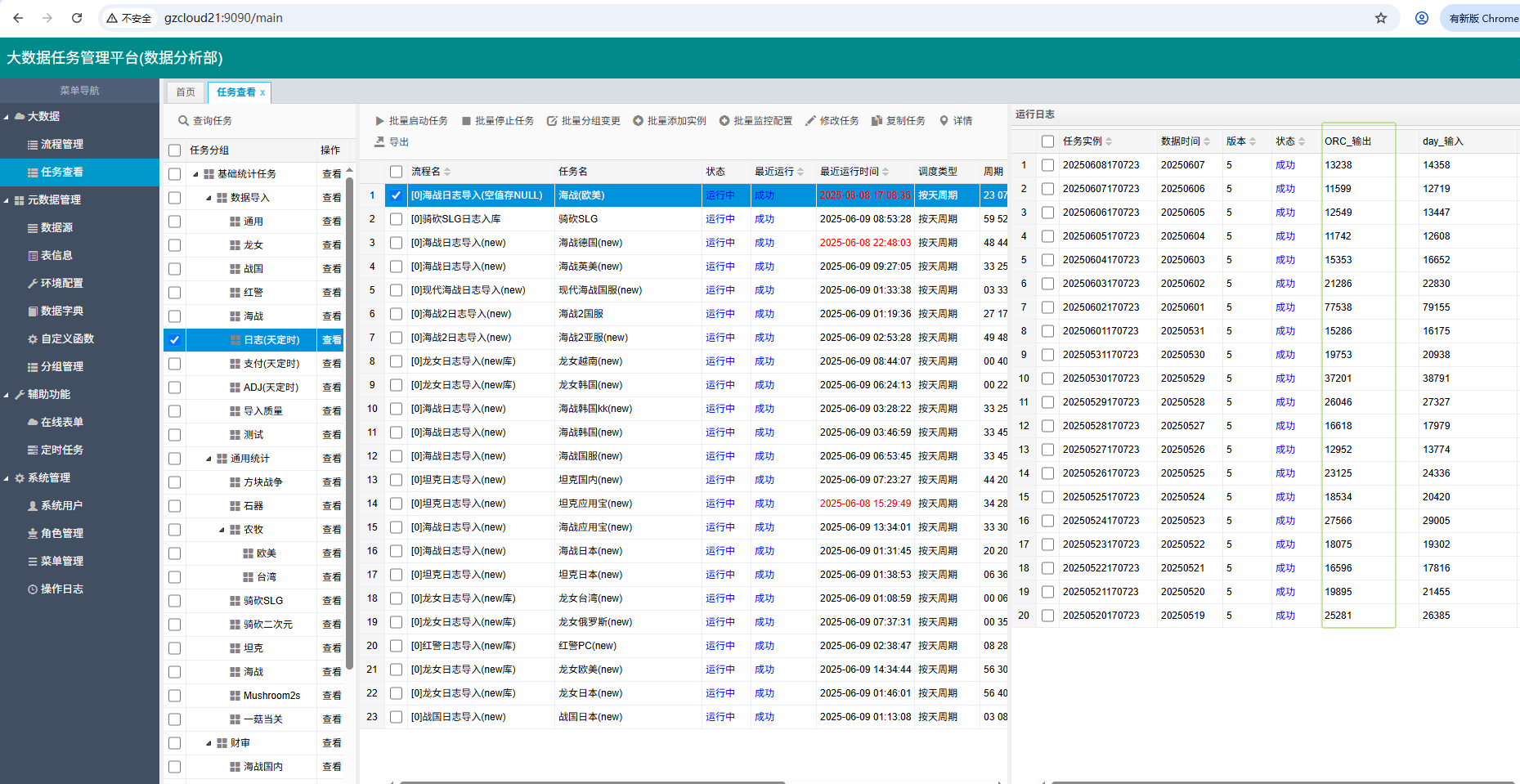
admin

shufen@#$%

# 定时任务汇总

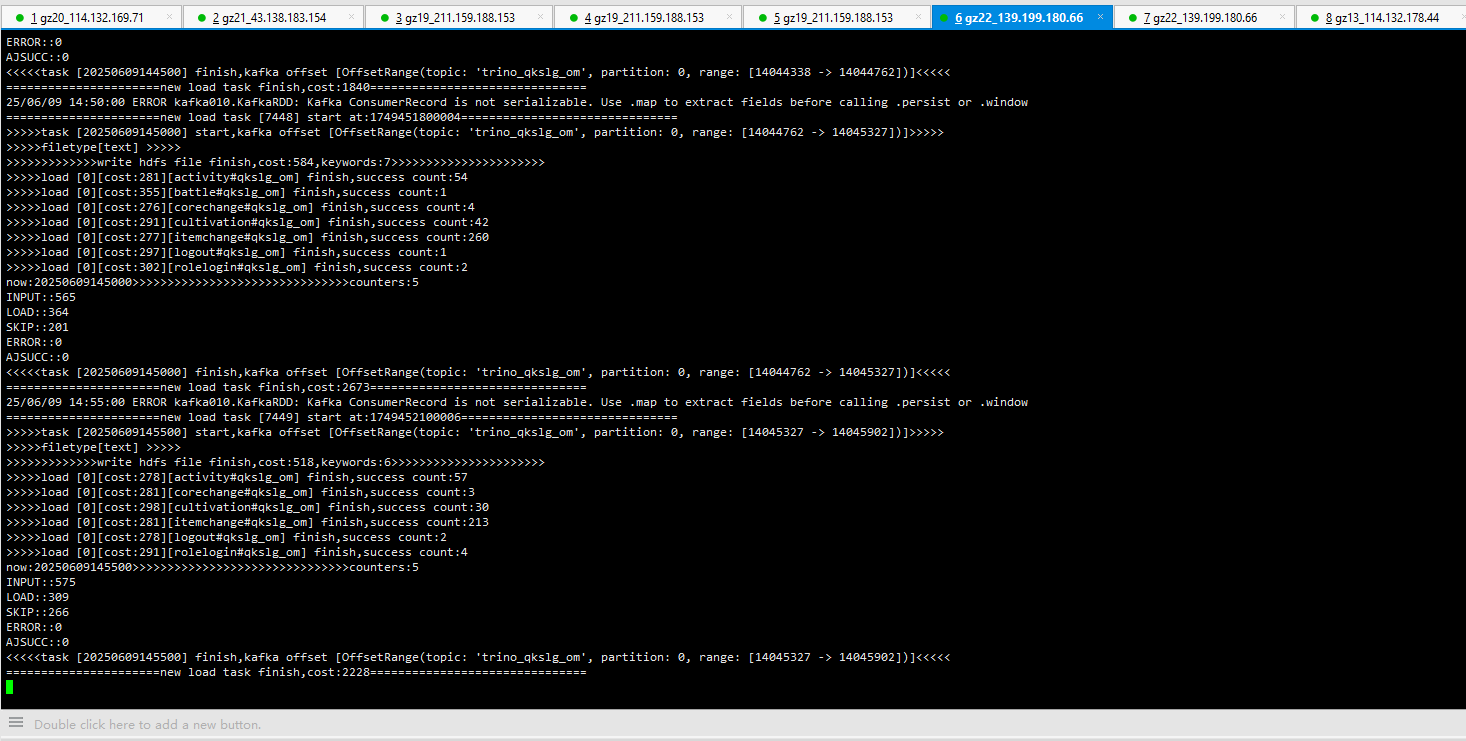


每天检查前一天的数据入库情况，如果发现输入为0，输出为0，先检查cos上相应的日志文件存不存在，再看yarn任务是否运行失败

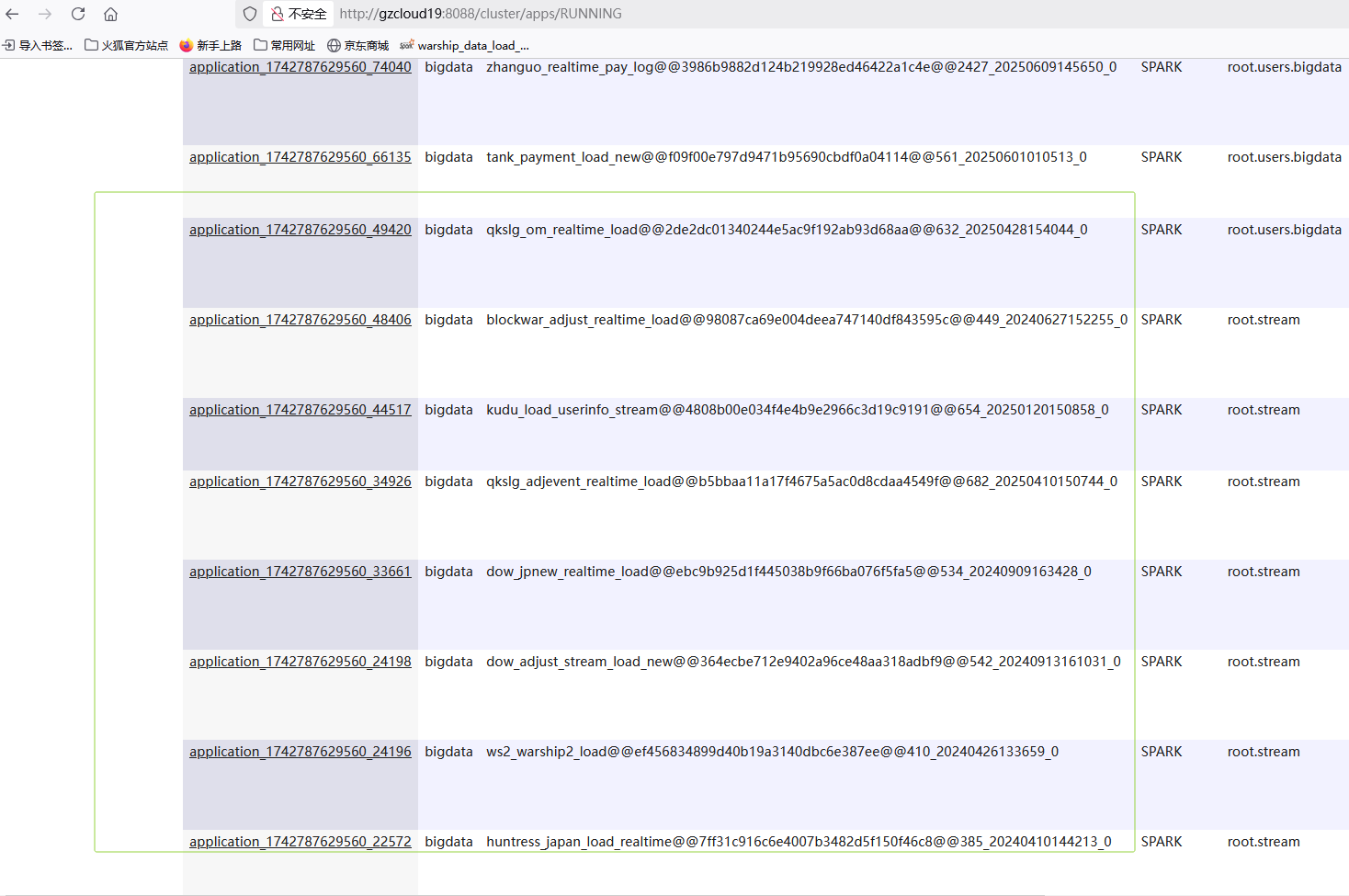


# 实时任务汇总

Shell上日志：



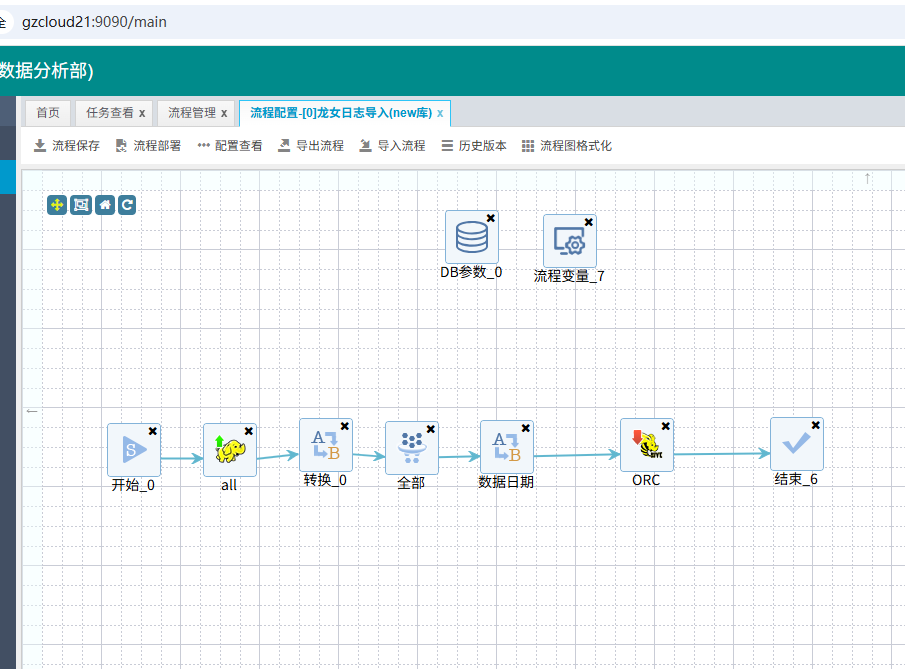
Yarn中任务



# 定时导入流程

以龙女日志导入为例

## 1.流程配置



日志读取

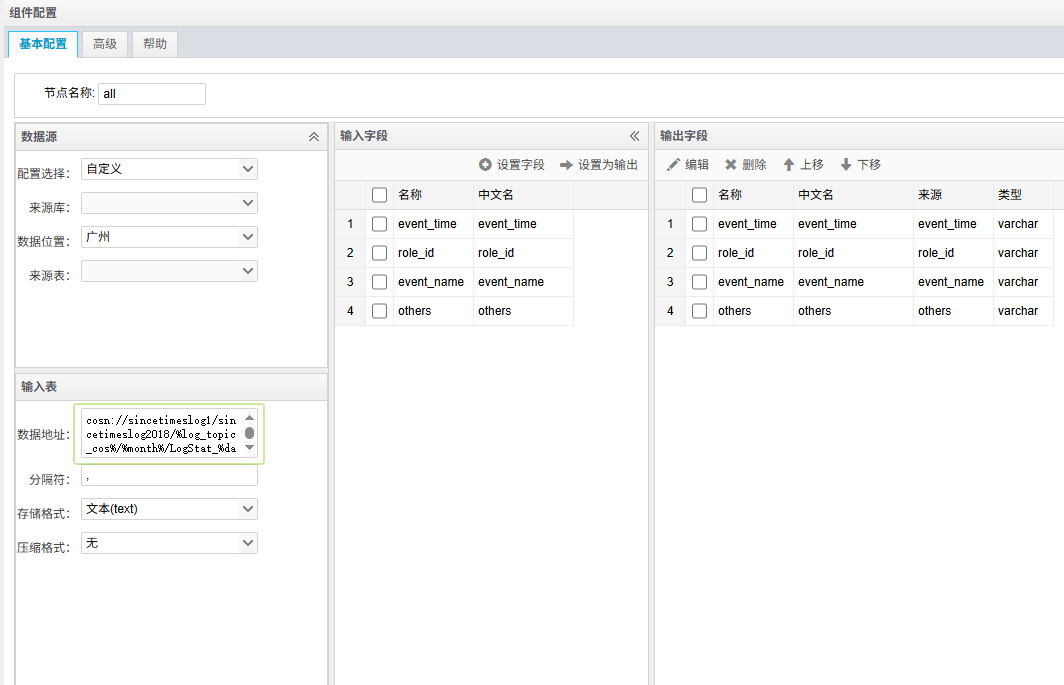
cosn://sincetimeslog1/sincetimeslog2018/%log\_topic\_cos%/%month%/LogStat\_%dayall%.log.tar.gz

cosn:开头表示直接读取cos日志文件

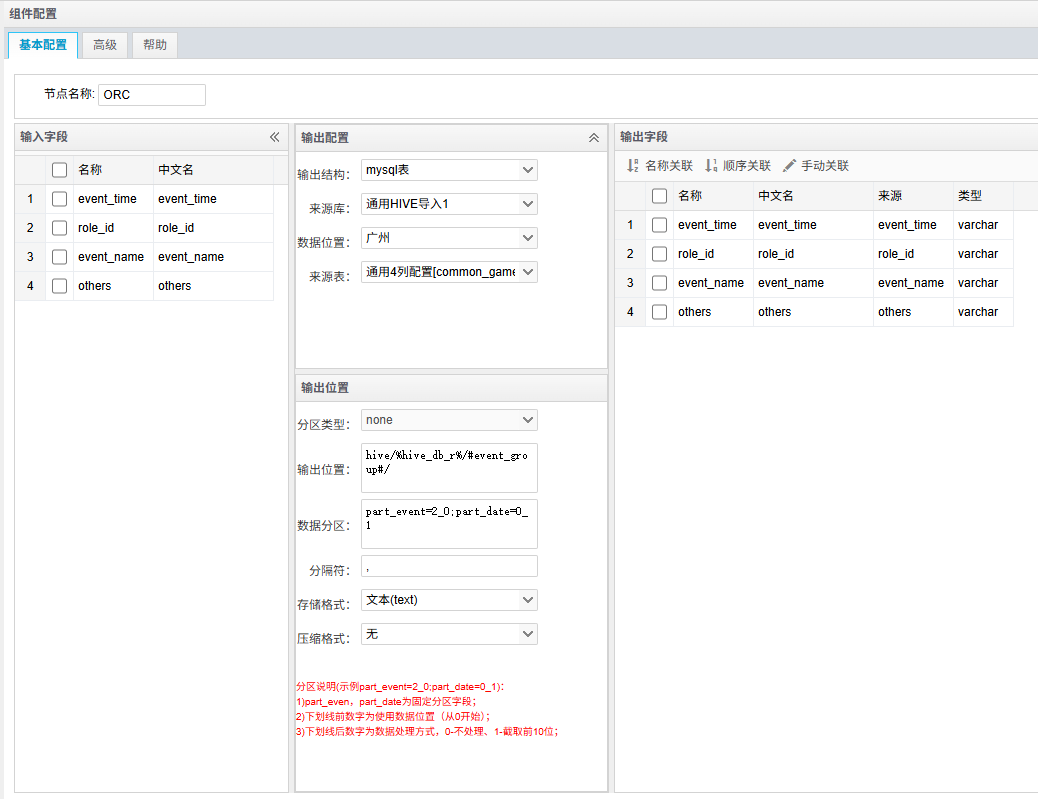
%log\_topic\_cos%：自定义参数，在增加任务时替换

%month%：系统内置参数，替换为执行任务的时间的 yyyyMM格式时间

%dayall%：系统内置参数，替换为执行任务的时间的 yyyy-MM-dd格式时间

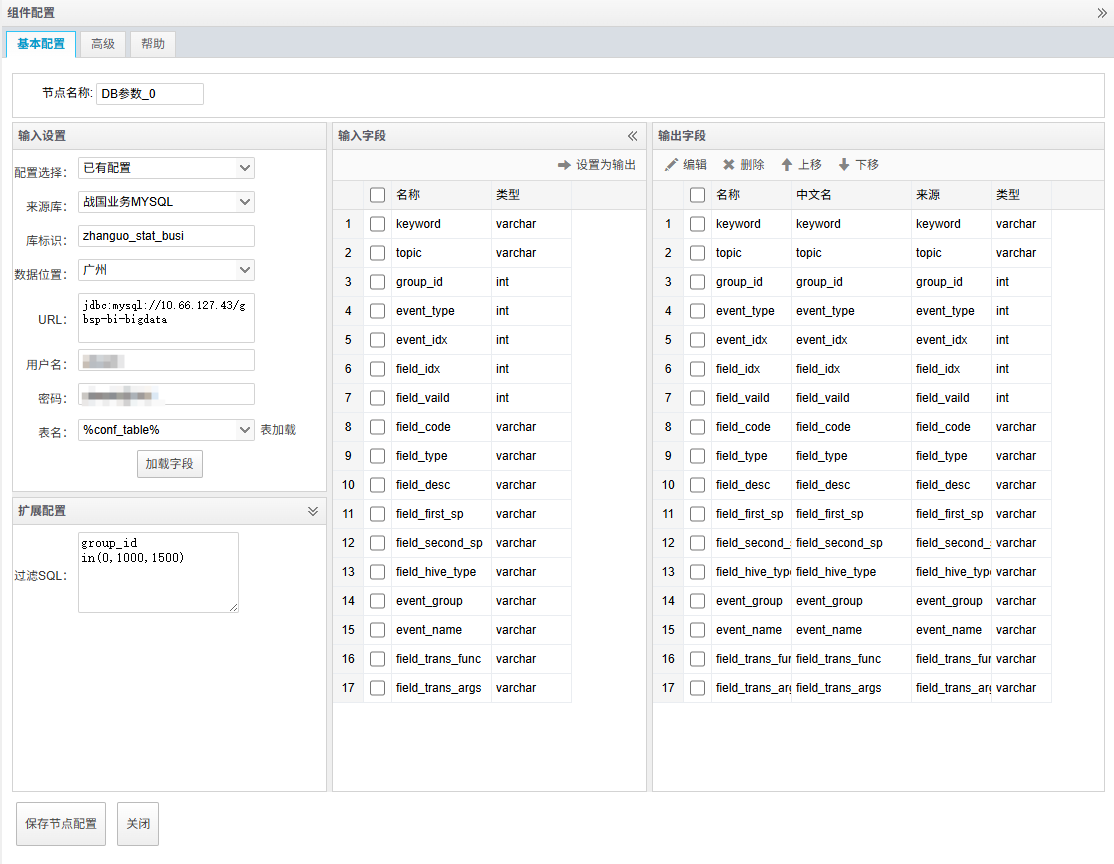


Hive日志输出

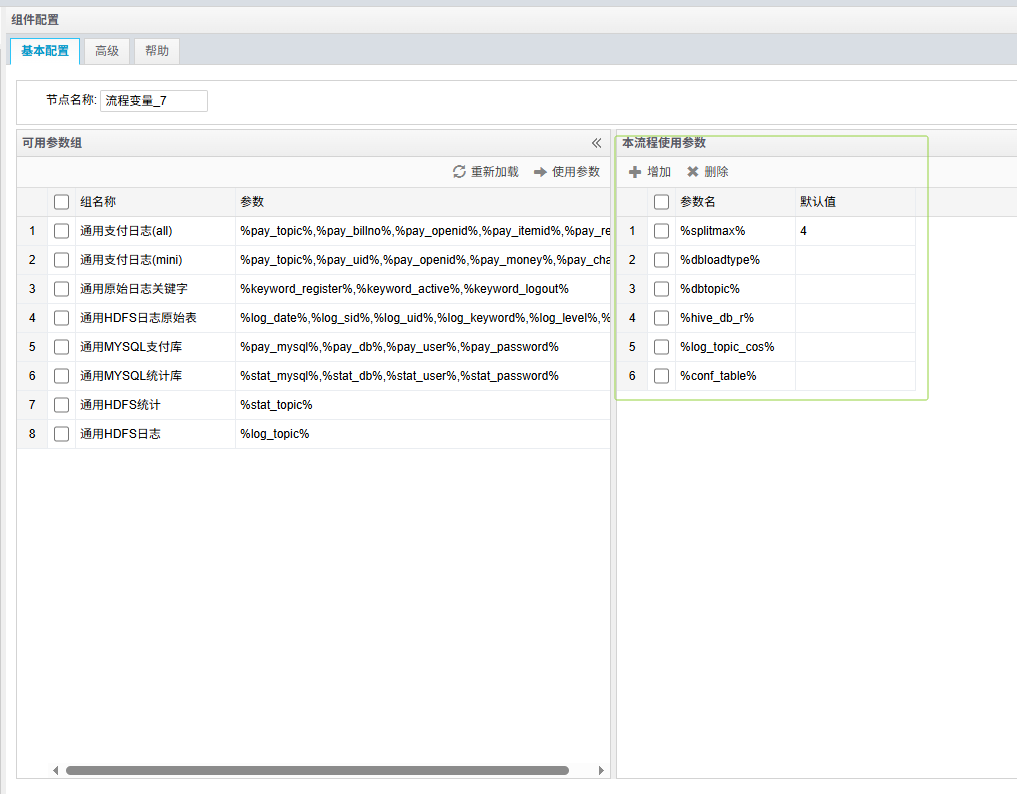


批量数据的输出位置一般为：huntress\_jpnew\_r

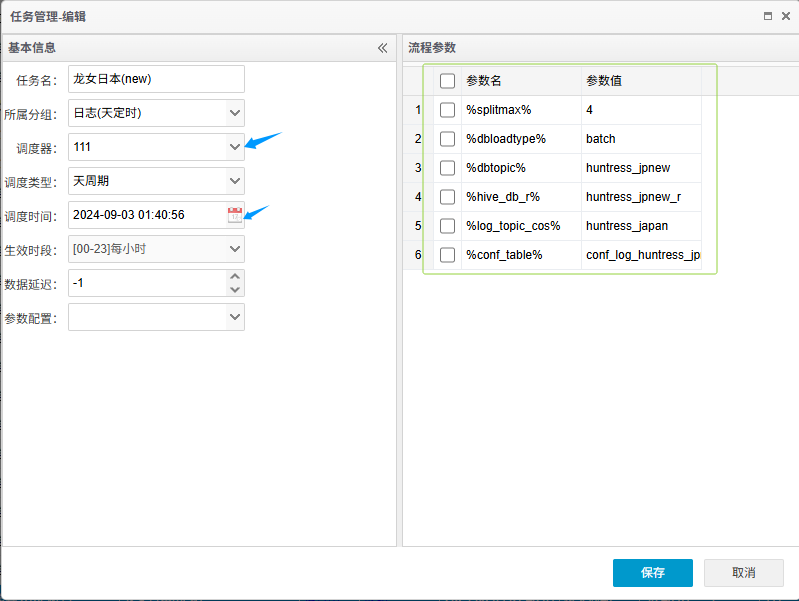
Trino入库配置表



流程自定义参数：



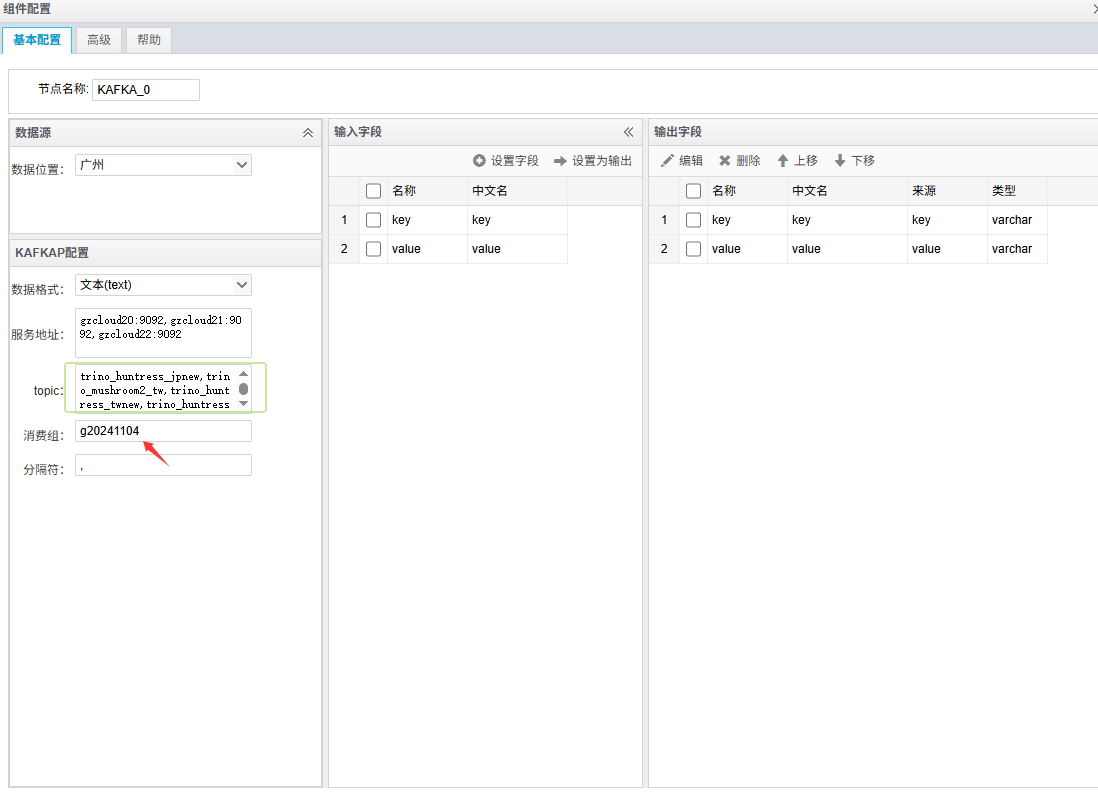
## 2.任务添加



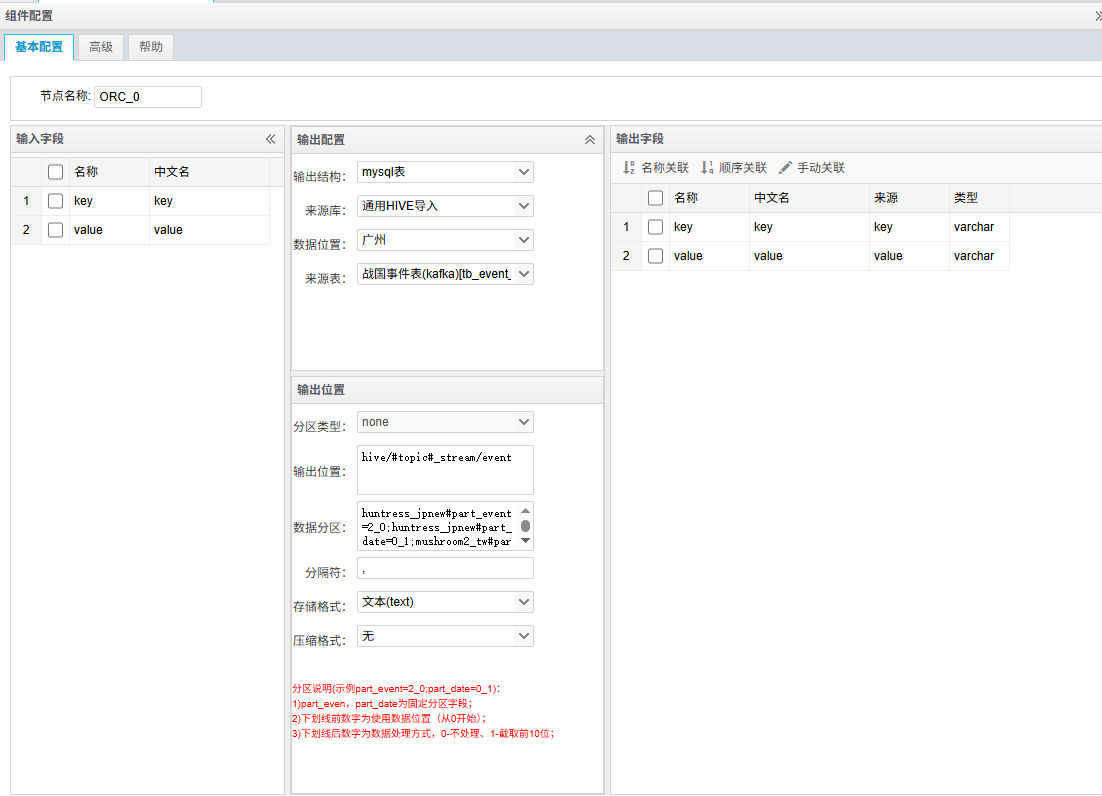
# 实时导入流程



Kafka读取，如果数据处理逻辑一致，可以同一个流程可以读取多个kafka的topic，注意下面的消费组，多个流程不能重复，现网流程的不能修改



输出组件



输出位置：实时数据的输出位置为 huntress\_jpnew**\_stream**

数据分区写法：part\_event和 part\_date必须成对出现

huntress\_jpnew#part\_event=2\_0;huntress\_jpnew#part\_date=0\_1;mushroom2\_tw#part\_event=1\_0;mushroom2\_tw#part\_date=0\_1

实时任务创建，在页面上可以任意创建一个任务，查看一下任务脚本，真实的实时任务并**不会使用页面创建的任务信息**

先查询页面任务脚本：

spark-submit --name huntress\_japan\_load\_realtime@@7ff31c916c6e4007b3482d5f150f46c8@@385\_20240410144213\_0 --conf spark.files.ignoreMissingFiles=true --master yarn --executor-memory 8G --driver-memory 4G --class com.sincetimes.bigdata.server.SparkProcessDriver /data/bigdata/flowhis/gbsp-processing-assembly-1.0.jar 10.104.13.152@@6379@@bigdata huntress\_japan\_load\_realtime@@7ff31c916c6e4007b3482d5f150f46c8@@385\_20240410144213\_0 20240430 '\*'

实时任务的真实脚本：

spark-submit --name huntress\_japan\_load\_realtime@@7ff31c916c6e4007b3482d5f150f46c8@@385\_20240410144213\_0 --conf spark.files.ignoreMissingFiles=true --master yarn --queue stream --class com.sincetimes.bigdata.server.SparkStreamingDriver /data/bigdata/mutilstream/gbsp-processing-assembly-3.0.jar 10.104.208.207@@6379@@bigdata huntress\_japan\_load\_realtime@@7ff31c916c6e4007b3482d5f150f46c8@@385\_20240410144213\_0 20240426 '\*'

gz19\_211.159.188.153：/data/bigdata/mutilstream/load\_huntress\_jp.sh

启动需要nohup的方式： nohup ./load\_huntress\_jp.sh > load.log &

注意：启动以后会一直在后台运行，也会一直占用yarn资源，注意实时任务不要启太多，否则会影响定时任务的录入

重启需要先kill掉原来的实时任务，注意先使用yarn命令kill任务，再使用linux的kill，实时任务一般在整5分钟运行数据录入，kill前先关注日志，不要在数据录入期间kill

