Flume+Kafka+Flink整合流程(所有组件已安装完毕)

启动服务进程

启动zookeeper

```
zkServer.sh start
```

启动kafka

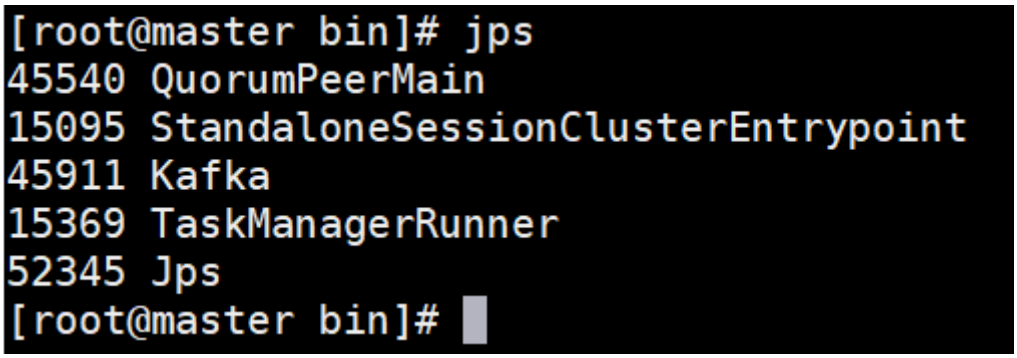
```
kafka-server-start.sh -daemon $KAFKA_HOME/config/server.properties
```

启动Flink

```
start-cluster.sh
```

查看启动进程是否成功

```
jps
```



注意：这里是单节点安装！！！

创建Kafka主题

创建 order 主题

```
kafka-topics.sh --create --zookeeper master:2181 --topic order --partitions 2 --replication-factor 1
```

查看 order 主题是否创建成功

```
kafka-topics.sh --list --zookeeper master:2181
```

准备数据生成脚本

在命令行中输入以下命令：

```
cd /opt
vi gen.sh
```

在文件中加入以下内容：

```
#!/bin/bash
function sendmsg(){
        exec 3<>/dev/"tcp"/master/26001
        num=0;
        while (($num < 1000))
        do
                echo "向端口发送数字$num"
                echo $num>&3
                (( num += 1 ))
                sleep 1
        done
        exec 3<&-
}
echo "---start---"
sendmsg
echo "----end----"
```

注意：在服务器实时计算任务的虚拟环境中已经存在数据生成脚本(socket_gen)，这里的脚本只是做演示！！！

准备Flume Agent配置文件

在命令行中输入以下命令：

```
cd $FLUME_HOME/conf
vi socket_kafka_simple.conf
```

在文件中加入以下内容：

```
# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = netcat
a1.sources.r1.bind = master
a1.sources.r1.port = 26001

# Describe the sink
# a1.sinks.k1.type = logger
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.topic =order
a1.sinks.k1.brokerList = master:9092
a1.sinks.k1.batchSize = 10
a1.sinks.k1.requiredAcks = 1
a1.sinks.k1.kafka.producer.linger.ms = 1
a1.sinks.k1.kafka.producer.compression.type = snappy

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 100
```

```
a1.channels.c1.transactionCapacity = 10

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

注意：这里的配置仅用于测试，具体情况根据题目设定！！！

测试端口数据是否能够传入kafkaTopic中

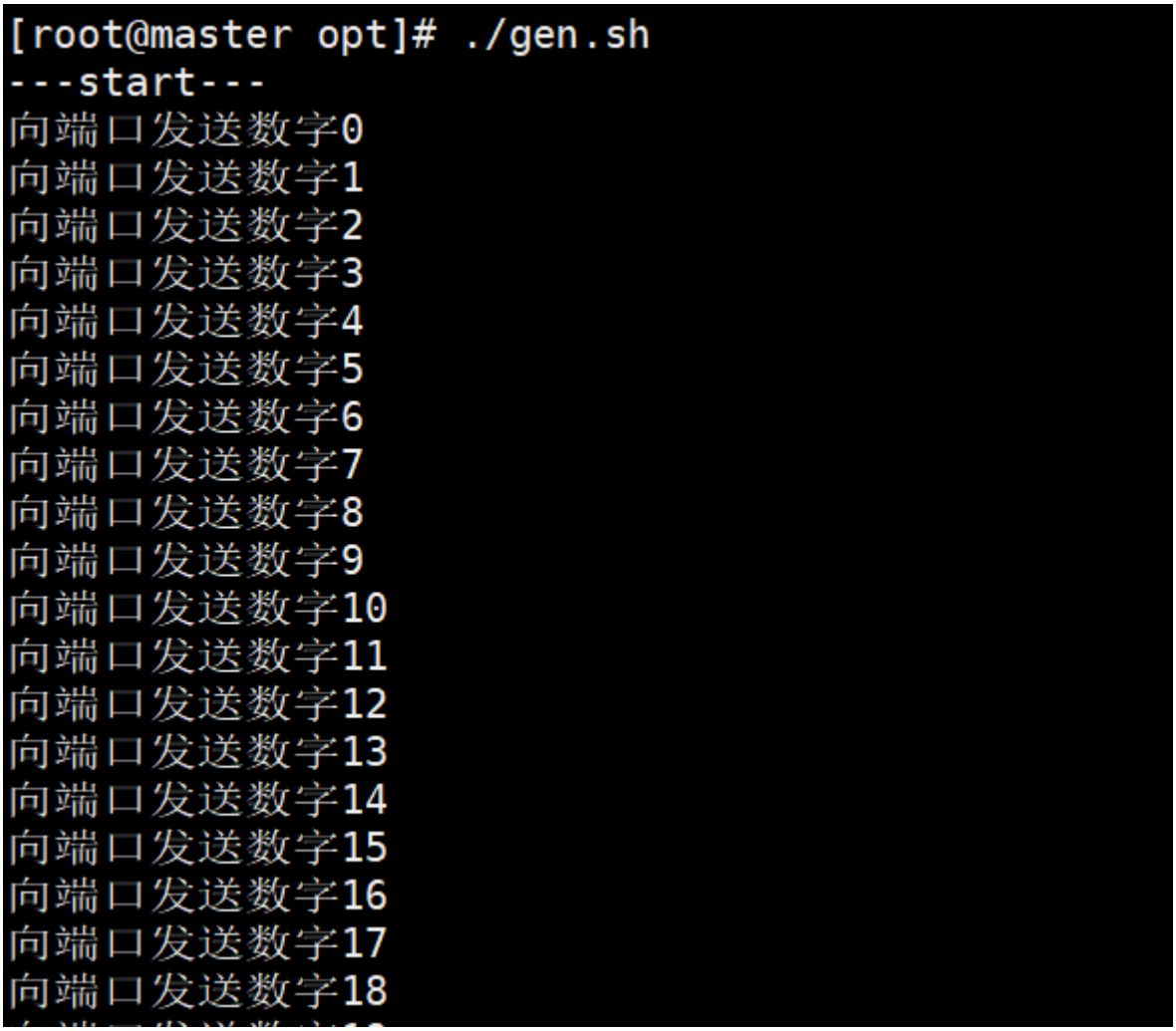
启动Flume监听端口数据

```
flume-ng agent --name a1 --conf-file $FLUME_HOME/conf/socket_kafka_simple.conf
```

注意：这里的Flume程序是作为服务端的，所以先启动Flume！！！

启动端口数据生成脚本

```
cd /opt
chmod +x gen.sh
./gen.sh
```



注意：这里的脚本由于是我们手动编写的，初始设置是没有执行权限的，因此需要赋予其可执行的权限！！！

检查flume Kafka Sink是否在生产数据

```
22/11/15 11:27:12 INFO producer.SyncProducer: Connected to master:9092 for
producing
22/11/15 11:27:12 INFO producer.SyncProducer: Disconnecting from master:9092
22/11/15 11:27:12 INFO producer.SyncProducer: Connected to 192.168.232.149:9092
for producing
```

查看kafkaTopic中实时数据

```
cd $KAFKA_HOME/logs/order-0
cat 00000000000000000000.log
```

注意：由于主题的分区数设定为2，会存在order-0和order-1两个数据文件夹！！！

测试Flink流处理程序消费kafkaTopic数据

编写Topic消费代码

```
package chapter05

import org.apache.flink.api.common.serialization.SimpleStringSchema
import org.apache.flink.streaming.api.scala._
import org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer

import java.util.Properties

object SourceKafkaTest {

  def main(args: Array[String]): Unit = {

    // 创建执行环境
    val env = StreamExecutionEnvironment.getExecutionEnvironment
    env.setParallelism(1)

    // 用properties保存Kafka连接的相关配置
    val properties = new Properties()

    // 这里的IP地址填服务器直连ip
    properties.setProperty("bootstrap.servers","192.168.232.149:9092")
    properties.setProperty("group.id","consumer-group")

    val stream:DataStream[String] = env.addSource(new FlinkKafkaConsumer[String]
("order", new SimpleStringSchema(), properties))

    stream.print()

    env.execute()
  }

}
```

注意：在运行该代码时，数据生成脚本和flume检控程序不要关闭！！！