/usr/local/apps/jdk1.8

Hadoop 2.6.5

Spark 2.2.0

zookeeper-3.4.12

apache-flume-1.7.0-bin

kafka\_2.11-1.1.0 2.10? kafka的包0.9不兼容0.8

hbase-2.0.1-bin

<http://spark.apache.org/docs/2.2.0/streaming-kafka-integration.html> 选择合适的spark-kafka连接包，不过其中有一个是过时的，很明显选stable版本

有telnet

java -cp Flume.data-1.0-SNAPSHOT.jar flume.test.data.FlumeTest

java -jar Flume.data-1.0-SNAPSHOT.jar flume.test.data.FlumeTest

nohup flume-ng agent -c /usr/local/apps/flume/conf -f /usr/local/apps/flume/conf/avro.conf -n a1 -Dflume.root.logger=INFO,console >/dev/null 2>&1 &

(hdp-node-01)

agent1.sources = source1  
agent1.sinks = sink1  
agent1.channels = channel1  
  
# Describe/configure source1  
agent1.sources.source1.type = avro  
agent1.sources.source1.bind = 0.0.0.0  
agent1.sources.source1.port = 4444  
  
agent1.sources.source1.channels=channel1  
agent1.channels.channel1.type=memory  
agent1.channels.channel1.capacity=2000  
  
agent1.channels.c1.transactionCapacity=200  
agent1.sinks.sink1.type= org.apache.flume.sink.kafka.KafkaSink  
agent1.sinks.sink1.channel=channel1

#

a1.sinks.k1.channel = channel1  
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink  
a1.sinks.k1.kafka.topic = test  
a1.sinks.k1.kafka.bootstrap.servers = hdp-node-01:9092

a1.sinks.k1.kafka.producer.compression.type = snappy

#关系不大的选项

#agent1.sinks.sink1.serializer.class=kafka.serializer.StringEncoder  
#a1.sinks.k1.kafka.flumeBatchSize = 20  
#a1.sinks.k1.kafka.producer.acks = 1  
#a1.sinks.k1.kafka.producer.linger.ms = 1

#无论是否选择producer.compression.type，kafka都会处理

#bootstrap.servers <-- broker.list

#avro Thrift等 RPC也是网络传输，但是客户端写起来... 肯定都是端口监听

Avro模式是使用JSON定义的

===============================================

nohup flume-ng agent -c /usr/local/apps/flume/conf -f /usr/local/apps/flume/conf/syslog\_tcp.conf -n a1 -Dflume.root.logger=INFO,console >/dev/null 2>&1 &

(hdp-node-02,hdp-node-03,hdp-node-04)

a1.sources = r1

a1.channels = c1

a1.sinks = k1

# Describe/configure the source

a1.sources.r1.type = syslogtcp

a1.sources.r1.port = 4444

a1.sources.r1.host = hdp-node-01

a1.sources.r1.channels = c1

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

# Describe the sink

a1.sinks.k1.channel = c1

a1.sinks.k1.type = org.apache.flume.sink.AvroSink

a1.sinks.k1.hostname = 192.168.27.101

a1.sinks.k1.port = 4444

===============================================

#zookeeper 的leader、follower 是谁并不重要

./bin/zkServer.sh status

./bin/zkServer.sh start

需要四个全部启动：

zookeeper必须比kafka先开

kafka-server-start.sh -daemon /usr/local/apps/kafka/config/server.properties

创建topic：

kafka-topics.sh --describe --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104 --topic test

kafka-topics.sh --create --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104 --replication-factor 2 --partitions 2 --topic test

删除topic：

kafka-topics.sh --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104:2181 --delete --topic test

kafka-topics.sh --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104:2181 --list

查看当前已有groupid：

kafka-consumer-groups.sh --list --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104 #但是我程序中的groupid不会被记住

kafka-consumer-groups.sh --list --bootstrap-server 192.168.27.101:9092 #无效

#group的作用 <https://blog.csdn.net/daiyutage/article/details/70599433>

对于kafka阻塞之后进行重启，原生脚本没用，需要自己kill

flume-ng 可能也需要重启

配置文件里面配置了zookeeper zookeeper.connect=192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104:2181：

kafka-console-producer.sh --broker-list 192.168.27.101:9092,192.168.27.102:9092,192.168.27.103:9092,192.168.27.104:9092 --topic test

kafka-console-consumer.sh --zookeeper 192.168.27.101:2181,192.168.27.102:2181,192.168.27.103:2181,192.168.27.104:2181 --topic test --from-beginning

#--from-beginning 会将服务器上存储的所有消息都传送过来，消息被消费之后会保留一定时间

#对于consumer而言,它需要保存消费消息的offset，已经消费到的地方。kafka集群几乎不需要维护任何consumer和producer状态信息,这些信息有zookeeper保存

./bin/start-hbase.sh

./bin/stop-hbase.sh