Laboratorio Kafka + Flume

Apartado 1. Arranque de Kafka

Levantamos unha máquina en OpenStack na que correr Kafka.

Editamos el **docker-compose.yaml,** co editor Nano poñendo como referencia puerto 9902 e a dirección IP de la máquina do CESGA:

```
---
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:7.3.0
    container_name: zookeeper
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000

  broker:
    image: confluentinc/cp-kafka:7.3.0
    container_name: broker
    ports:
      - "9092:9092"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_INTERNAL:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://10.133.28.84:9092,PLAINTEXT_INTERNAL://broker:29092
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
```


```
---
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:7.3.0
    container_name: zookeeper
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000

  broker:
    image: confluentinc/cp-kafka:7.3.0
    container_name: broker
    ports:
      - "9092:9092"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_INTERNAL:PLAINTEXT
```


Agora crearemos un topic para este laboratorio, para iso antes deberemos facer un docjer compose up -d:

```
docker compose up -d
```

```
docker exec broker \
kafka-topics --bootstrap-server broker:9092 \
             --create \
             --topic laboratorio
```

```
cesgaxuser@xuwira02:~$ docker exec broker kafka-topics --bootstrap-server broker:9092 --create --topic
 laboratorio
Error response from daemon: No such container: broker
cesgaxuser@xuwira02:~$ docker compose up -d
[+] Running 16/16
 ✔ zookeeper 2 layers [⣿⣿]      0B/0B      Pulled                                                  15.7s
   ✔ fa08a06f385f Pull complete                                                                   5.3s
   ✔ bddb49e2fc4d Pull complete                                                                   5.2s
 ✔ broker 12 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled                                          15.5s
   ✔ d5d2e87c6892 Pull complete                                                                   1.0s
   ✔ 008dba906bf6 Pull complete                                                                   0.4s
   ✔ bfeaabe01655 Pull complete                                                                   4.1s
   ✔ 2cb7eb0f5666 Pull complete                                                                   1.0s
   ✔ f70f416c6ce7 Pull complete                                                                   1.3s
   ✔ bc67d000e59b Pull complete                                                                   2.0s
   ✔ d6e744651f37 Pull complete                                                                   1.7s
   ✔ 0427d86fae81 Pull complete                                                                   2.1s
   ✔ 4108e73e61e1 Pull complete                                                                   2.4s
   ✔ ac5563423559 Pull complete                                                                   2.5s
   ✔ d32323e291f3 Pull complete                                                                   4.9s
   ✔ ee69ff430d89 Pull complete                                                                   2.9s
[+] Running 2/3
 ⠿ Network cesgaxuser_default  Created                                                            3.9s
 ✔ Container zookeeper         Started                                                          3.3s
 ✔ Container broker            Started                                                          1.0s
cesgaxuser@xuwira02:~$ docker exec broker kafka-topics --bootstrap-server broker:9092 --create --topic
 laboratorio
Created topic laboratorio.
cesgaxuser@xuwira02:~$
```

Apartado 2. Configuración do axente Flume a consola

Crearemos agora o nano **axente-kafka.conf:**

```
# Define un memory channel chamado ch1 en axentekafka
axentekafka.channels.ch1.type = memory

# Define un source de tipo kafka
# Indica o ip e porto ao que conectarse para consumir os datos de kafka
axentekafka.sources.kafka-source1.type = org.apache.flume.source.kafka.KafkaSource
axentekafka.sources.kafka-source1.kafka.bootstrap.servers = 10.133.28.84:9092
axentekafka.sources.kafka-source1.kafka.topics = laboratorio
axentekafka.sources.kafka-source1.batchSize = 100
axentekafka.sources.kafka-source1.channels = ch1

# Define un sink de tipo log
axentekafka.sinks.log-sink1.channel = ch1
axentekafka.sinks.log-sink1.type = logger

# Indica ao axente axentekafka que componhentes activar
axentekafka.channels = ch1
axentekafka.sources = kafka-source1
axentekafka.sinks = log-sink1
```

```
# Define un memory channel chamado ch1 en axentekafka
axentekafka.channels.ch1.type = memory

# Define un source de tipo kafka
# Indica o ip e porto ao que conectarse para consumir os datos de kafka
axentekafka.sources.kafka-source1.type = org.apache.flume.source.kafka.KafkaS
axentekafka.sources.kafka-source1.kafka.bootstrap.servers = 10.133.28.84:9092
axentekafka.sources.kafka-source1.kafka.topics = laboratorio
axentekafka.sources.kafka-source1.batchSize = 100
axentekafka.sources.kafka-source1.channels = ch1

# Define un sink de tipo log
axentekafka.sinks.log-sink1.channel = ch1
axentekafka.sinks.log-sink1.type = logger

# Indica ao axente axentekafka que componhentes activar
axentekafka.channels = ch1
axentekafka.sources = kafka-source1
axentekafka.sinks = log-sink1
```

Levantamos o axente na máquina hadoop,así o axente Flume estaría agora "subscrito" como "consumer" a Kafka, no topic "laboratorio":

```
[xuwira40@cdh61-login4 ~]$ flume-ng agent --conf ./flume/conf/ -f flume/conf/axente-kafka.conf -n
axentekafka
```

```
[xuwira02@cdh61-login5 ~]$ flume-ng agent --conf ./flume/conf/ -f flume/conf/axe
nte-kafka.conf -n axentekafka
Info: Including Hadoop libraries found via (/bin/hadoop) for HDFS access
Info: Including HBASE libraries found via (/bin/hbase) for HBASE access
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated a
nd will likely be removed in a future release
Error: Could not find or load main class org.apache.hadoop.hbase.util.GetJavaPro
perty
Info: Including Hive libraries found via () for Hive access
+ exec /usr/java/jdk1.8.0_191-amd64/bin/java -Xmx20m -cp '/home/xunta/wir/a02/fl
ume/conf:/opt/cloudera/parcels/CDH-6.1.1-1.cdh6.1.1.p0.875250/lib/flume-ng/lib/*
```

Agora utilizaremos o producer e escribiremos elementos no topic "laboratorio", e comprobaremos no flume se está sendo recibido ao poñer a escoitar:

```
docker exec --interactive --tty broker \
kafka-console-producer --bootstrap-server broker:9092 \
                       --topic laboratorio
```

```
cesgaxuser@xuwira02:~$ docker exec --interactive --tty broker \
> kafka-console-producer --bootstrap-server broker:9092 \
>                        --topic laboratorio
>hola
>como
>estamos
>todos?
```

```
24/01/29 21:48:55 INFO internals.Fetcher: [Consumer clientId=consumer-1, groupId
=flume] Resetting offset for partition laboratorio-0 to offset 0.
24/01/29 21:50:17 INFO sink.LoggerSink: Event: { headers:{topic=laboratorio, par
tition=0, timestamp=1706561416507} body: 68 6F 6C 61 20
        hola  }
24/01/29 21:50:17 INFO sink.LoggerSink: Event: { headers:{topic=laboratorio, par
tition=0, timestamp=1706561416667} body: 63 6F 6D 6F 20
        como  }
24/01/29 21:50:21 INFO sink.LoggerSink: Event: { headers:{topic=laboratorio, par
tition=0, timestamp=1706561419501} body: 65 73 74 61 6D 6F 73 20
        estamos  }
24/01/29 21:50:53 INFO sink.LoggerSink: Event: { headers:{topic=laboratorio, par
tition=0, timestamp=1706561452516} body: 74 6F 64 6F 73 3F
        todos? }
```

Apartado 3. Productor con Node-Red

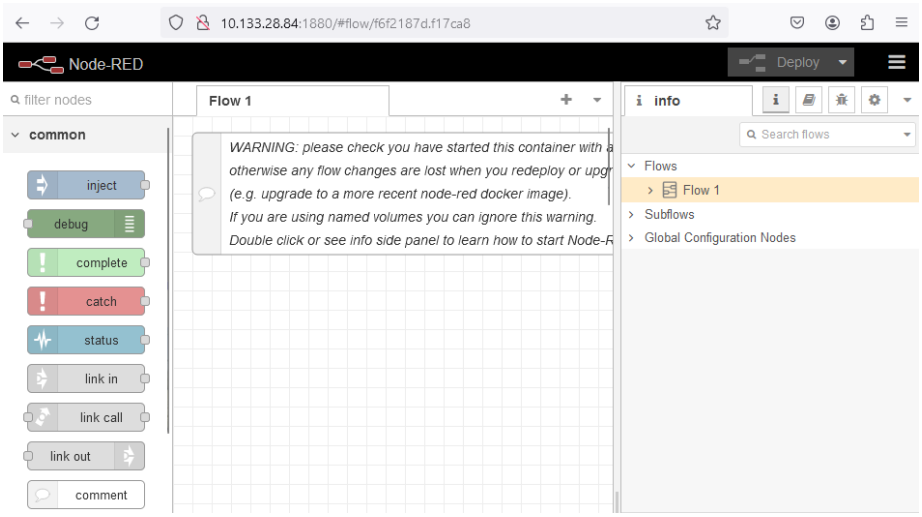
Levantamos un contedor de Node-Red dende a mesma máquina que corre kafka.

```
docker run -it -p 1880:1880 -v node_red_data:/data --name mynodered -d nodered/node-red
```

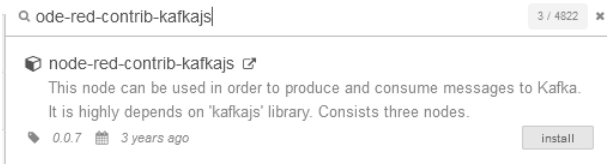
```
cesgaxuser@xuwira02:~$ docker run -it -p 1880:1880 -v node_red_data:/data --name mynodered -d nodered/
node-red
Unable to find image 'nodered/node-red:latest' locally
latest: Pulling from nodered/node-red
7264a8db6415: Pull complete
eee371b9ce3f: Pull complete
93b3025fe103: Pull complete
d9059661ce70: Pull complete
485fe505b563: Pull complete
033625439e2c: Pull complete
d4cf1bf4e6f5: Pull complete
4f4fb700ef54: Pull complete
21b718421b0f: Pull complete
31d890393399: Pull complete
2e1de13dbff0: Pull complete
b2a15cef18bc: Pull complete
a1accdfcafe7: Extracting     832B/832B
bb06094bcfa4: Download complete
6495b004c211: Download complete
ac475ea8b303: Downloading  65.96MB/116.7MB
131fca43d2af: Download complete
```

Node-Red está executandose na máquina remota en OpenStack, dispoñible no porto 1880. Deberemos pegar a nosa IP co porto no navegador web:

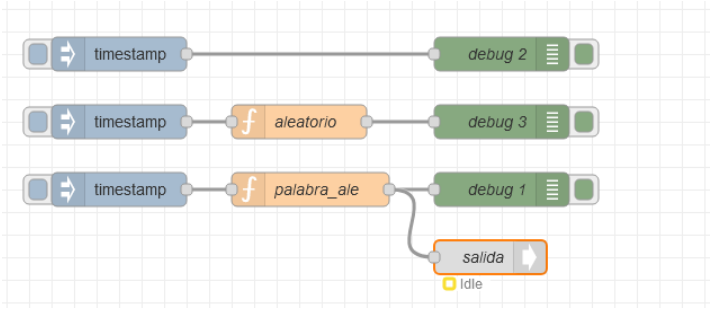
```
ssh -L 1880:10.133.28.84:1880 xuwira02@hadoop.cesga.es
```



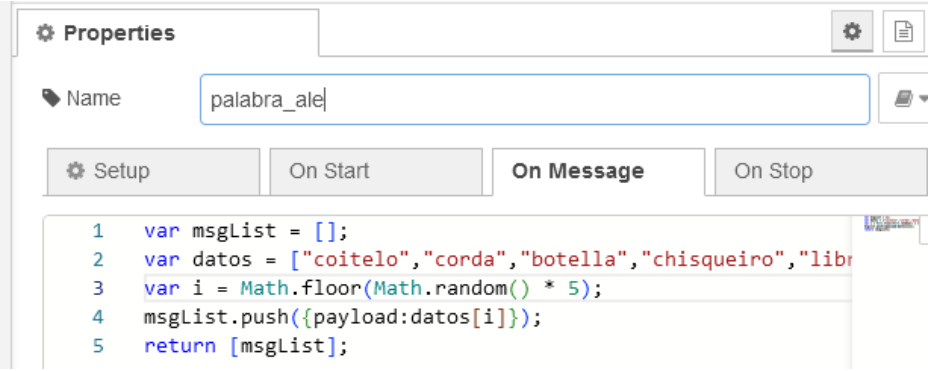
Instalación do plugin para traballo con Kafka:
Opcións -> Manage Palette -> Install -> node-red-contrib-kafkajs



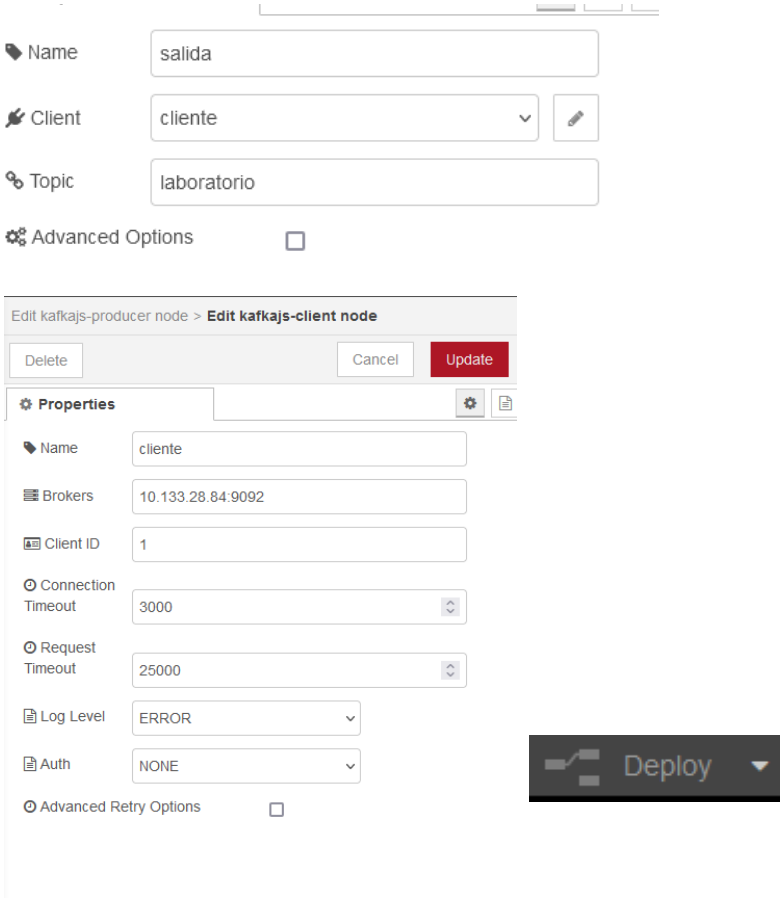
Agora crearemos o seguinte esqueleto

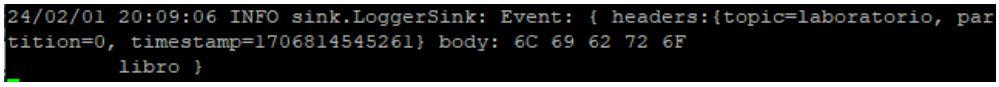


Dentro das funcións pegaremos os distintos anexos:



Na salida, deberemos de introducir a etiqueta,de topic laboratorio e o cliente que será co porto de kafka:

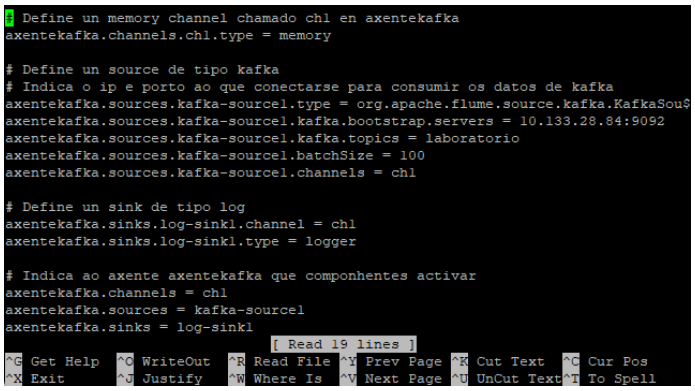




Apartado 4. Configuración do axente Flume a HDFS

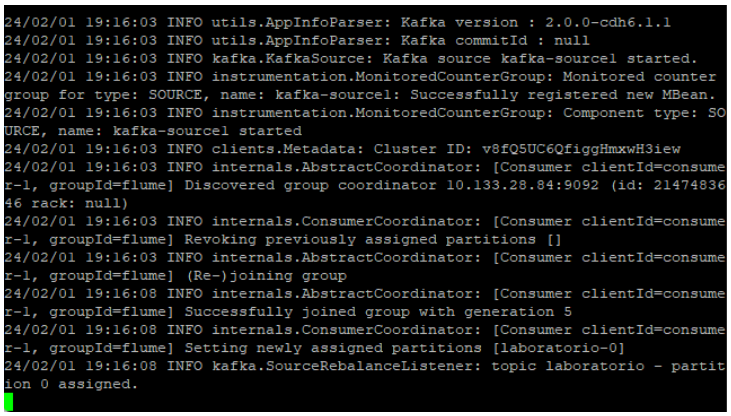
Agora que temos node-red producindo datos e enviandoos a Kafka é hora de volcar os datos en HDFS, o noso obxectivo principal. Crearemos por tanto o **axente-kafka-hdfs.conf**:

```
# Define un memory channel chamado ch1 en axentekafka
axentekafka.channels.ch1.type = memory

# Define un source de tipo kafka
# Indica o ip e porto ao que conectarse para consumir os datos de kafka
axentekafka.sources.kafka-source1.type = org.apache.flume.source.kafka.KafkaSource
axentekafka.sources.kafka-source1.kafka.bootstrap.servers = 10.133.28.84:9092
axentekafka.sources.kafka-source1.kafka.topics = laboratorio
axentekafka.sources.kafka-source1.batchSize = 100
axentekafka.sources.kafka-source1.channels = ch1

# Define un sink de tipo hdfs
axentekafka.sinks.log-sink1.channel = ch1
axentekafka.sinks.log-sink1.type = hdfs
axentekafka.sinks.log-sink1.hdfs.path = hdfs://nameservice1/user/xuwira02/kafkeando


# Indica ao axente axentekafka que componhentes activar
axentekafka.channels = ch1
axentekafka.sources = kafka-source1
axentekafka.sinks = log-sink1
```

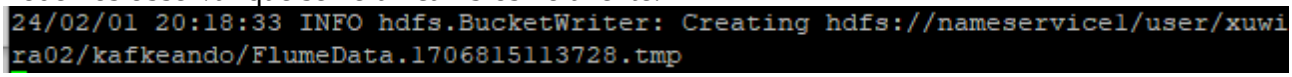


Levantamos o axente:

```
[xuwira40@cdh61-login4 ~]$ flume-ng agent --conf ./flume/conf/ -f flume/conf/axente-kafka-hdfs.conf
-n axentekafka
```

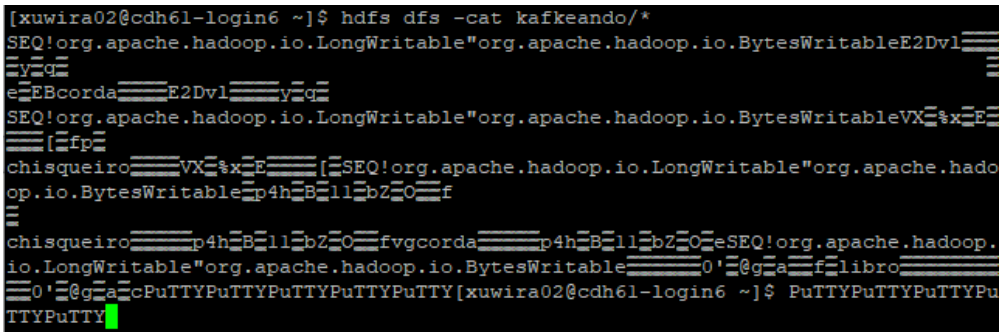


Podemos observar que se xeran cambios no axente:



Utiliza Node-Red para producir mensaxes e verifica a continuación se se escribiron en HDFS.

```
[xuwira40@cdh61-login4 ~]$ hdfs dfs -cat kafkeando/*
```



Podemos observar que si se escribiron correctamente dentro do HDFS.