|  |  |
| --- | --- |
| Alumno | Araceli Macía Barrado |
|  | Ejercicio Final Individual - Extractores de Datos |

# Ejercicio 1

En este ejercicio vamos a utilizar Flume para consumir eventos syslog y almacenarlos en HDFS:



Fichero de configuración: /etc/flume/conf/syslog-to-hdfs.properties

agent1.sources = source1

agent1.sinks = sink1

agent1.channels = channel1

#configuro el channel entre source y sink

agent1.sources.source1.channels = channel1

agent1.sinks.sink1.channel = channel1

#configuracion del source

agent1.sources.source1.type = syslogudp

agent1.sources.source1.port = 5140

agent1.sources.source1.host = localhost

#configuracion de sink

agent1.sinks.sink1.type = hdfs

#especificamos la ruta destino en hdfs:

agent1.sinks.sink1.hdfs.path = /tmp/flume/syslog

#prefijo para los ficheros que cree flume

agent1.sinks.sink1.hdfs.filePrefix = events

#prefijo para los ficheros que estan siendo escritos:

agent1.sinks.sink1.hdfs.inUsePrefix = \_

#tipo de fichero de salida, en este caso queremos texto plano:

agent1.sinks.sink1.hdfs.fileType = DataStream

#Configuracion del channel.

agent1.channels.channel1.type = file

Comandos utilizados:

* Ejecucion del agente:

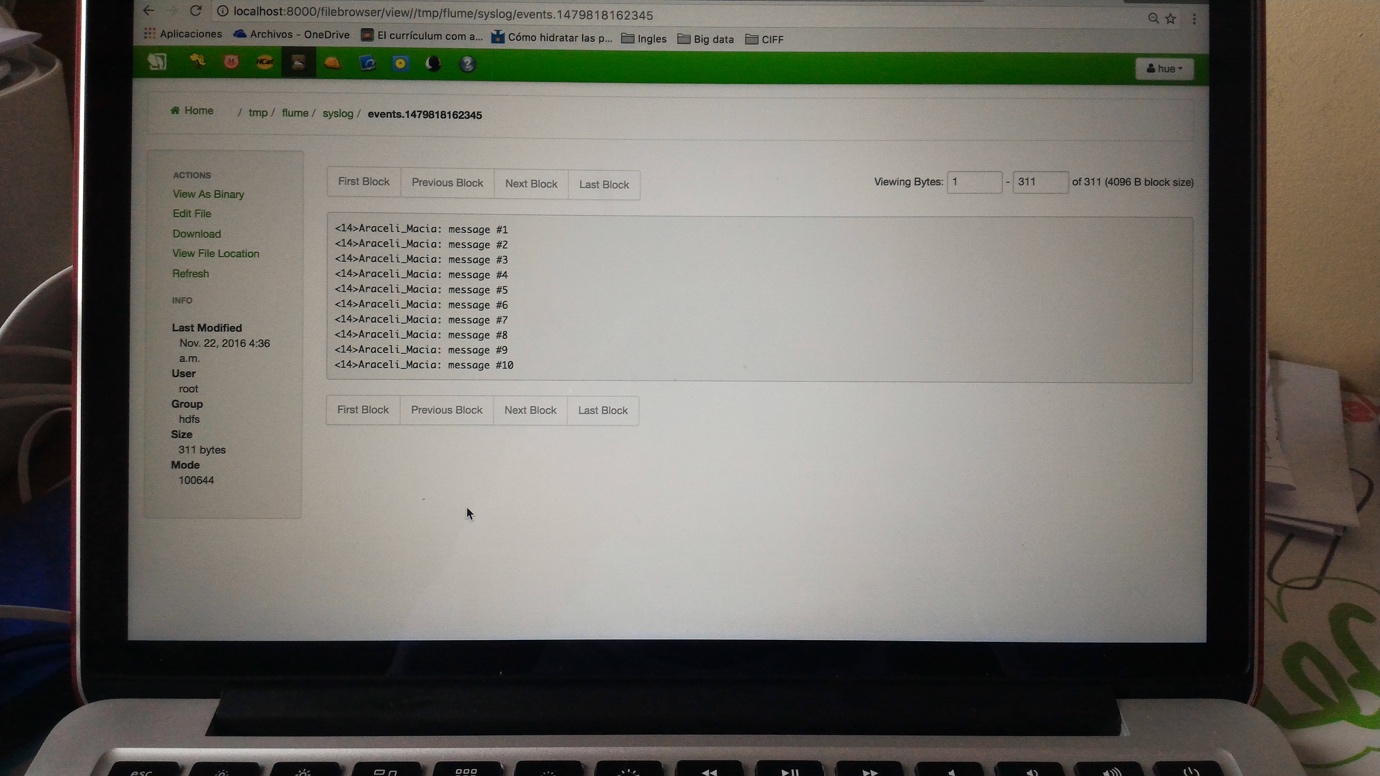
flume-ng agent --conf-file /etc/flume/conf/syslog-to-hdfs.properties --name agent1 --conf /etc/flume/conf/ -Dflume.root.logger=INFO,console

* Ejecucion de script de generacion de eventos:

python syslog\_generator.py Araceli\_Macia

Foto del resultado Final:

Foto de uno de los fichero generados que tienen mi nombre:



# Ejercicio 2

1. Inicia un cluster de Kafka que consista en 3 brokers (recuerda que tendrás que usar un fichero de configuración diferente para cada uno, al igual que hicimos en los ejercicios de clase)
2. Crea un topic “practica” que tenga 3 particiones y factor de replicación 3
3. Usa la opción --describe del comando kafka-topics.sh para comprobar que el tópico practica que acabas de crear tiene 3 particiones
4. Usando el console-producer que incluye Kafka (y que usamos en algunos ejercicios de clase), inserta algunos mensajes en el topic, asegúrate de que uno de los mensajes sea tu nombre y apellidos. Una vez publicados los mensajes, ya puedes matar al console-producer. (Usando las teclas Ctrl+C)
5. Usando el console-consumer que incluye Kafka, consume **todos los mensajes** del topic practica
6. Mata uno de los brokers (Usando Ctrl+C)
7. Vuelve a lanzar el console-consumer y consume todos los mensajes del topic practica
8. Mata otro broker (Usando Ctrl+C)
9. Vuelve a lanzar el console-consumer y consume todos los mensajes del topic practica

* Fichero de configuracion para iniciar un cluster de Kafka con 3 brokers

He creado 2 ficheros de configuración, uno para cada broker:

* Directorio : /usr/hdp/current/kafka-broker/config

Ficheros con lineas modificadas:

* + server-2.properties
    - port=6668
    - log.dirs=/kafka-logs2
    - broker.id=1
  + server-3.properties
    - port=6669
    - log.dirs=/kafka-logs3
    - broker.id=2
* Iniciar el cluster, inicio los 3 brokers, cada uno en un terminal.

cd /usr/hdp/current/kafka-broker/

bin/kafka-server-start.sh config/server.properties (en terminal 1)

bin/kafka-server-start.sh config/server-2.properties (en terminal 2)

bin/kafka-server-start.sh config/server-3.properties (en terminal 3)

* Crea un topic “practica” que tenga 3 particiones y factor de replicación 3

cd /usr/hdp/current/kafka-broker/

bin/kafka-topics.sh --create --zookeeper localhost:2181 --topic PRACTICA --partitions 3 --replication-factor 3

* Usa la opción --describe del comando kafka-topics.sh para comprobar que el tópico practica que acabas de crear tiene 3 particiones

cd /usr/hdp/current/kafka-broker/

bin/kafka-topics.sh --zookeeper localhost:2181 --describe

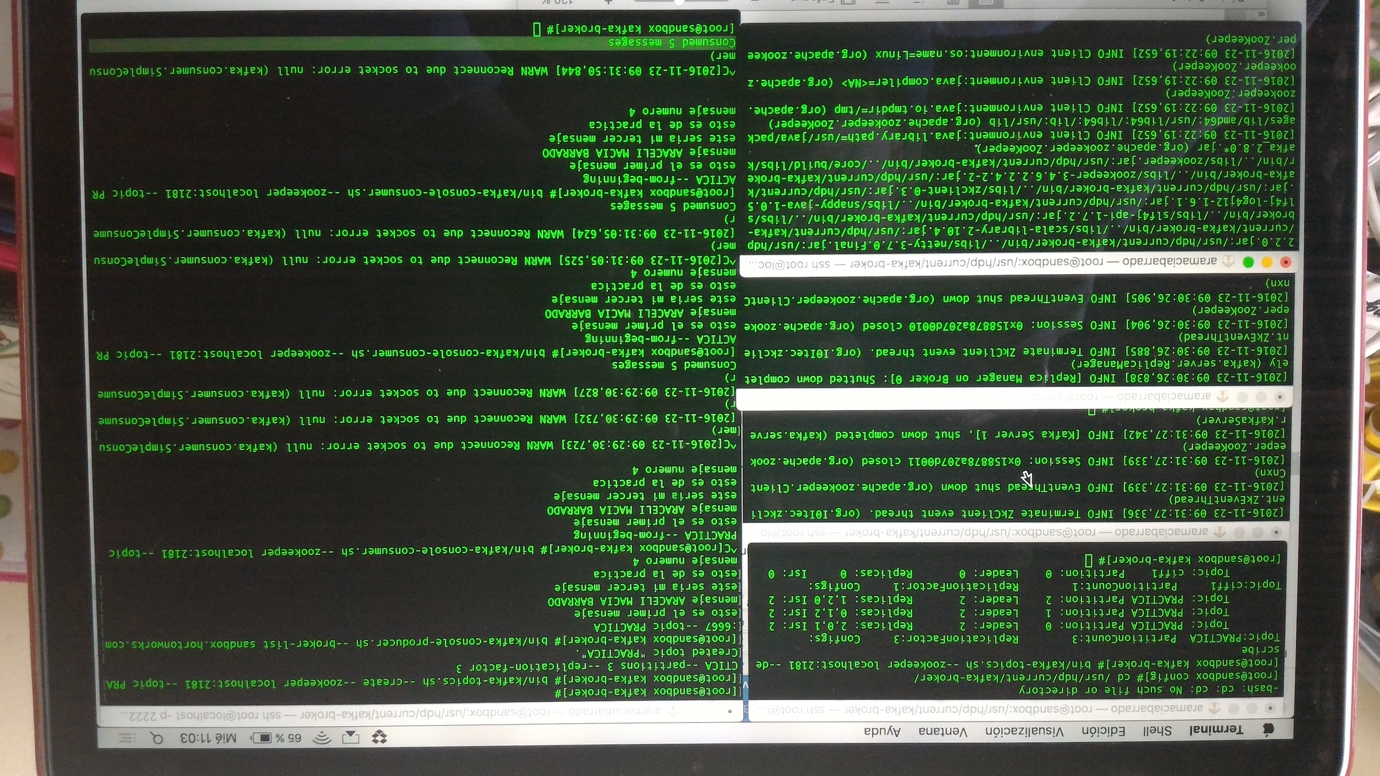
* Lanzar el console-producer

bin/kafka-console-producer.sh --broker-list sandbox.hortonworks.com:6667 --topic PRACTICA

* Lanzar el console-consumer

bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic PRACTICA --from-beginning

* FOTO donde se ve que, aunque se hayan “Caído” dos brokers, se puede seguir consumiendo sin pérdida de datos.



# Ejercicio 3

El ejercicio consiste en ejecutar una importación de Sqoop que cree en HDFS un dataset de empleados “veteranos” en el directorio /sqoop/veteran\_employees.

Debes ejecutar una importación de sqoop (sqoop import) similar a las de los ejercicios de clase, pero no debes importar todos los datos de la tabla, sino sólo aquellos cuya fecha de contratación (hire\_date) sea menor que el 1 de Enero de 1990.

* **Comando Utilizado:**

sqoop import --connect jdbc:mysql://localhost/employees --table employees --where " hire\_date < '1990-01-01'" --fields-terminated-by "\t" --warehouse-dir /sqoop/veteran\_employees

* **Foto:**

