Práctica 3: Captura de datos con Flume

Para capturar datos con Flume es necesario configurar un agente que recibe datos desde una fuente (source), que volcará en un sumidero (sink).

Hay que crear un archivo que contiene las configuraciones necesarias de:

• Los componentes del agente: indican los nombres del source, sink y channel.
• La configuración del source: en este caso se abre una conexión con la aplicación netcat en el puerto 55555
• La configuración del channel: tipo (por memoria) y capacidades del canal para los eventos
• La configuración del sink: el destino son archivos almacenados en hdfs

Las línea del archivo **agenteFlume.conf** son, en este caso

```
#declaracion de componentes
agente.sources = sr1
agente.channels = chn1
agente.sinks = snk1
#configuracion del source
agente.sources.sr1.type = netcat
agente.sources.sr1.bind = localhost
agente.sources.sr1.port = 55555
agente.sources.sr1.channels = chn1
#configuracion del channel
agente.channels.chn1.type = memory
#La cantidad maxima de eventos almacenados en el canal
agente.channels.chn1.capacity = 1000
#La cantidad maxima de eventos que el canal capturara de la fuente por transacción
agente.channels.chn1.transactionCapacity = 100
#definimos la configuracion del Sink
agente.sinks.snk1.type = hdfs
agente.sinks.snk1.hdfs.path = hdfs://node1:8020/user/alumno/flume-puerto
#DataStream no comprimira el archivo de salida
agente.sinks.snk1.hdfs.fileType = DataStream
agente.sinks.snk1.channel = chn1
```

NOTA: Definimos el source, channel y sink (fuente, canal, sumidero).
NOTA: Hemos utilizado como fuente el netcat escuchando en el puerto 55555 y el canal memoria, por último el sumidero que es el sistema de ficheros hdfs.

Arranque del proceso de Flume

Ejecutar el siguiente comando para lanzar el proceso Flume para ejecutar el archivo creado. Se usa la opción –conf-file para indicar el archivo de configuración, -conf para la configuración de Flume, y -name para el nombre del agente (debe coincidir con el del fichero de configuración agenteFlume.conf)

sudo flume-ng agent --conf /etc/flume-ng/conf --conf-file agenteFlume.conf \
-name agente -Dflume.root.logger=INFO,console

NOTA: El comando se tiene que ejecutar sin backslash.

```
[alumno@pasarela flume]$ sudo flume-ng agent --conf /etc/flume-ng/conf/ --conf-file agenteFlume.conf --n
ame agente -Dflume.root.logger=INFO,console
[sudo] password for alumno:
Info: Including Hadoop libraries found via (/bin/hadoop) for HDFS access
Info: Including HBASE libraries found via (/bin/hbase) for HBASE access
Error: no se ha encontrado o cargado la clase principal org.apache.hadoop.hbase.util.GetJavaProperty
Info: Including Hive libraries found via () for Hive access
```

--conf → indica el directorio donde se cargará el archivo de configuración del agente.
--conf-file → indica el archivo de configuración para el agente.
--name → indica el nombre del agente para identificarlo.
--Dflume.root.logger=INFO,console → indica que la salida de logs será por la consola.

Debe aparecer en la última línea este mensaje que indica que Flume se ha conectado al puerto ....Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[/127.0.0.1:55555]

```
2024-02-21 19:33:06,025 (lifecycleSupervisor-1-2) [INFO - org.apache.flume.source.NetcatSource.start(Net
catSource.java:166)] Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[/127.0.0.1:55555]
```

Arranque de la fuente de datos

Como en este caso el origen del canal de Flume es la aplicación netcat, lanzamos este comando para enviar datos al agente Flume y escribimos alguna frase de ejemplo:

“nc localhost 55555 [ENTER]” & “<Texto_a_insertar> [ENTER]”

```
[alumno@pasarela flume]$ nc localhost 55555
Muestra de la escritura de este ordenador
OK
```

Y el terminal donde lanzamos el proceso Flume, debería aparecer en la última línea un mensaje parecido a este:

Creating hdfs://node1:8020/user/alumno/flume-puerto/FlumeData.163XXXXX444

```
fs.BucketWriter.open(BucketWriter.java:246)] Creating hdfs://node1:8020/user/alumno/flume-puerto/FlumeDa
ta.1708540920275.tmp
2024-02-21 19:42:31,513 (hdfs-snk1-roll-timer-0) [INFO - org.apache.flume.sink.hdfs.HDFSEventSink$1.run(
HDFSEventSink.java:381)] Writer callback called.
2024-02-21 19:42:31,514 (hdfs-snk1-roll-timer-0) [INFO - org.apache.flume.sink.hdfs.BucketWriter.doClose
(BucketWriter.java:438)] Closing hdfs://node1:8020/user/alumno/flume-puerto/FlumeData.1708540920275.tmp
2024-02-21 19:42:31,548 (hdfs-snk1-call-runner-4) [INFO - org.apache.flume.sink.hdfs.BucketWriter$7.call
(BucketWriter.java:681)] Renaming hdfs://node1:8020/user/alumno/flume-puerto/FlumeData.1708540920275.tmp
_to hdfs://node1:8020/user/alumno/flume-puerto/FlumeData.1708540920275
```

Para comprobar que el puerto está escuchando podemos usar el comando “lsof -i -P -n”

```
[alumno@pasarela flume]$ lsof -i -P -n | grep 55555
nc         8908 alumno    4u  IPv4 330550      0t0  TCP 127.0.0.1:48248->127.0.0.1:55555 (ESTABLISHED)
```

Comprobación de los datos escritos en HDFS

Se puede verificar que se están escribiendo los datos en HDFS en la ruta configurada como salida /user/alumno/flume-puerto/

Podemos utilizar el comando “hadoop fs -ls <ruta_hdfs>”

```
[alumno@pasarela flume]$ hadoop fs -ls /user/alumno/flume-puerto/
Found 1 items
-rw-r--r--   2 root supergroup         42 2024-02-21 19:42 /user/alumno/flume-puerto/FlumeData.170854092
0275
```

Recoge en un pantallazo os datos escritos con Flume en HDFS como muestra de que has hecho la práctica. Recuerda que se debe ver tu nombre en la imagen.

```
[alumno@pasarela flume]$ FRANCISCO GARRIDO LARA^C
[alumno@pasarela flume]$ hadoop fs -cat /user/alumno/flume-puerto/FlumeData.1708540920275
Muestra de la escritura de este ordenador
[alumno@pasarela flume]$
```