Apache Flume Tutorial : Twitter Data Streaming

https://www.edureka.co/blog/apache-flume-tutorial/

En este tutorial de Apache Flume, entenderemos cómo Flume ayuda en la transmisión de datos de varias fuentes. Pero antes de eso, entendamos la importancia de la ingesta de datos. La ingesta de datos es el paso inicial e importante para procesar y analizar datos, y luego derivar los valores de negocios de ella. Existen múltiples fuentes de datos que se recopilan en una organización.

Flume se hizo tan popular. Espero que pueda estar familiarizado con Apache Hadoop, que se está utilizando ampliamente en la industria. Flume puede integrarse fácilmente con Hadoop y descargar datos no estructurados y semiestructurados en HDFS, complementando el poder de Hadoop. Es por esto que Apache Flume es una parte importante del ecosistema Hadoop.

En este tutorial del blog de Apache Flume, cubriremos:

- Introducción a Apache Flume
- Ventajas de Apache Flume
- Arquitectura del canal
- Transmisión de datos de Twitter utilizando Flume

Comenzaremos este tutorial de Flume discutiendo sobre lo que es Apache Flume. Luego, avanzando, entenderemos las ventajas de usar Flume.

Introducción a Apache Flume

Apache Flume es una herramienta para la ingestión de datos en HDFS. Recopila, agrega y transporta gran cantidad de datos de transmisión, como archivos de registro, eventos de diversas fuentes, como tráfico de red, redes sociales, mensajes de correo electrónico, etc. a HDFS. Flume es una herramienta altamente confiable y distribuida.

La idea principal detrás del diseño de Flume es capturar datos de transmisión desde varios servidores web a HDFS. Tiene una arquitectura simple y flexible basada en flujos de datos de transmisión. Es tolerante a fallos y proporciona un mecanismo de fiabilidad para la tolerancia a fallos y la recuperación de fallos.

Ventajas de Apache Flume

Hay varias ventajas de Apache Flume que lo convierten en una mejor opción sobre los demás. Las ventajas son:

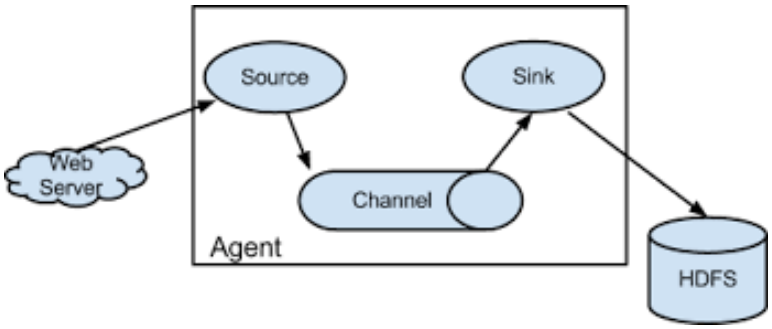
- Flume es escalable, confiable, tolerante a fallas y personalizable para diferentes fuentes (sources) y Sinks.
- Apache Flume puede almacenar datos en almacenamientos centralizadas (es decir, los datos provienen de un solo almacenamiento) como HBase y HDFS.
- Flume es horizontalmente escalable.
- Si la velocidad de lectura excede la velocidad de escritura, Flume proporciona un flujo constante de datos entre las operaciones de lectura y escritura.

- Flume proporciona la entrega de mensajes confiable. Las transacciones en Flume se basan en el canal donde se mantienen dos transacciones (un remitente y un receptor) para cada mensaje.
- Usando Flume, podemos ingerir datos de múltiples servidores en Hadoop.
- Nos brinda una solución confiable y distribuida y nos ayuda a recopilar, agregar y mover una gran cantidad de conjuntos de datos como Facebook, Twitter y sitios web de comercio electrónico.
- Nos ayuda a ingerir datos de transmisión en línea de varias fuentes como tráfico de red, redes sociales, mensajes de correo electrónico, archivos de registro, etc. en HDFS.
- Es compatible con un gran conjunto de fuentes y tipos de destinos.

La arquitectura es una de las que permite a Apache Flume obtener estos beneficios. Ahora, como conocemos las ventajas de Apache Flume, avancemos y entendamos la arquitectura de Apache Flume.

Arquitectura de Flume

Ahora, entendamos la arquitectura de Flume a partir del siguiente diagrama:

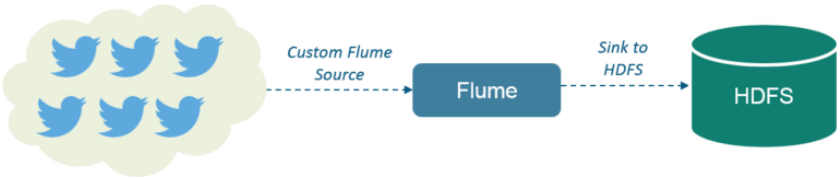


Hay un agente de Flume que ingiere los datos de transmisión de una o varias fuentes de datos a HDFS. Desde el diagrama, puede comprender fácilmente que el servidor web indica la fuente de datos. Twitter es una de las fuentes famosas de transmisión de datos.

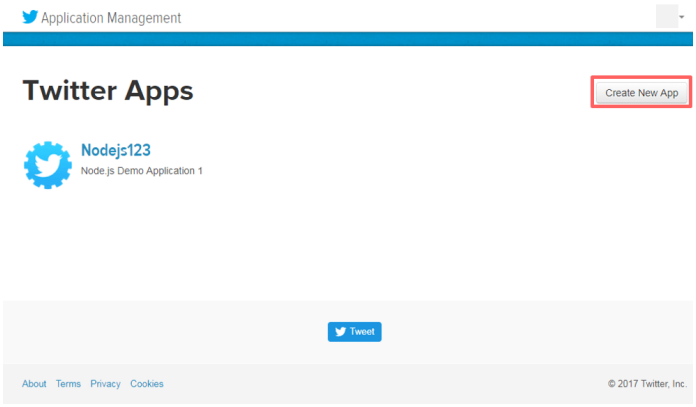
El agente del canal tiene 3 componentes: fuente (Source), sumidero (Sink) y canal (Channel).

Transmisión de datos de Twitter

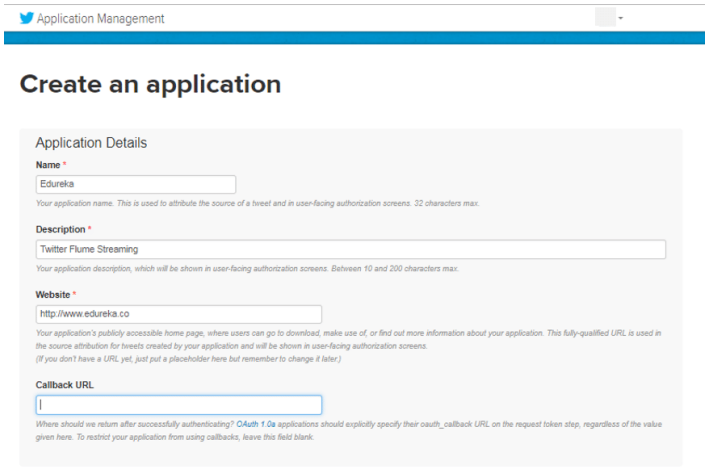
En esta práctica, transmitiremos datos de Twitter con Flume y luego almacenaremos los datos en HDFS como se muestra en la imagen de abajo.



El primer paso es crear una aplicación de Twitter. Para esto, primero tienes que ir a esta url: https://apps.twitter.com/ e iniciar sesión en tu cuenta de Twitter. Vaya a crear pestaña de aplicación como se muestra en la imagen de abajo.

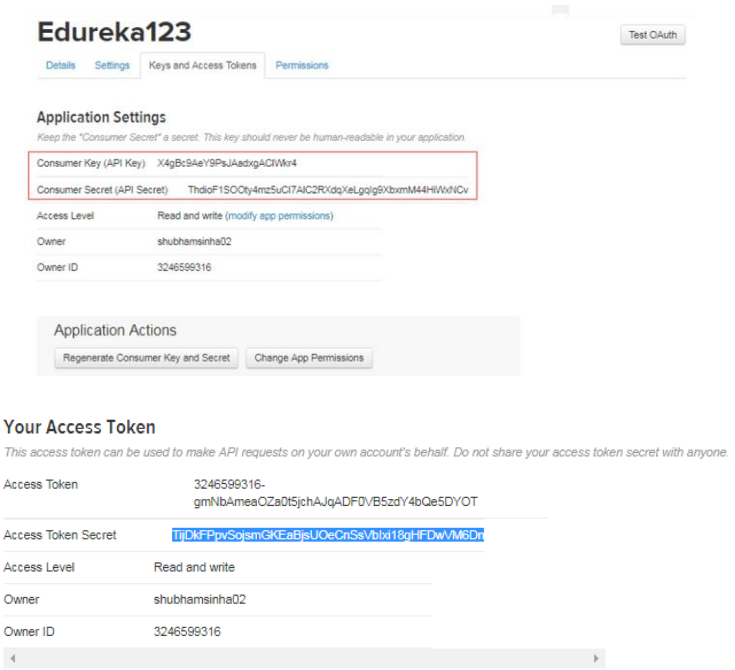


Luego, cree una aplicación como se muestra en la imagen de abajo.



Después de crear esta aplicación, encontrará el token de clave y acceso (Key & Access token). Copia la clave y el token de acceso.

Pasaremos estos tokens en nuestro archivo de configuración de Flume para conectarnos a esta aplicación.



Ahora cree un archivo `flume.conf` en el directorio raíz del canal como se muestra en la siguiente imagen. Como vimos, en la Arquitectura de Flume, configuraremos nuestra Fuente, Sink y Canal.

Nuestra fuente es Twitter, desde donde estamos transmitiendo los datos y nuestra Sink es HDFS, donde estamos escribiendo los datos.

En la configuración de origen de datos, estamos pasando el tipo de Twitter como `org.apache.flume.source.twitter.TwitterSource`. Entonces, indicaremos los cuatro tokens que recibimos de Twitter, para tener acceso a la información

Para finalizar, en la configuración de origen de datos, estamos indicando las palabras clave(keywords) por las que vamos a buscar los tweets.

En la configuración de Sink vamos a configurar las propiedades de HDFS. Estableceremos la ruta HDFS, el formato de escritura, el tipo de archivo, el tamaño del lote, etc. Finalmente, vamos a configurar el canal de memoria como se muestra en la imagen de abajo.

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS

TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = X4gBc9AeY9PsJAadxgAClWkr4
TwitterAgent.sources.Twitter.consumerSecret =  ThdioF1SOOty4mz5uCI7AlC2RXdqXeLgqIg9XbxmM44HiWxNCv
TwitterAgent.sources.Twitter.accessToken = 3246599316-gmNbAmeaOZa0t5jchAJqADF0VB5zdY4bQe5DYOT
TwitterAgent.sources.Twitter.accessTokenSecret = TijDkFPpvSojsmGKEaBjsUOeCnSsVbIxi18gHFDwVM6Dn
TwitterAgent.sources.Twitter.keywords = hadoop, big data, analytics, bigdata, cloudera, data science, data scientist, business intelligence,
mapreduce, data warehouse, data warehousing, mahout, hbase, nosql, newsql, businessintelligence, cloudcomputing

TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/user/flume/tweets/
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000
TwitterAgent.sinks.HDFS.hdfs.rollInterval = 600

TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 100
```

Source Properties

Sink Properties

Channel Properties

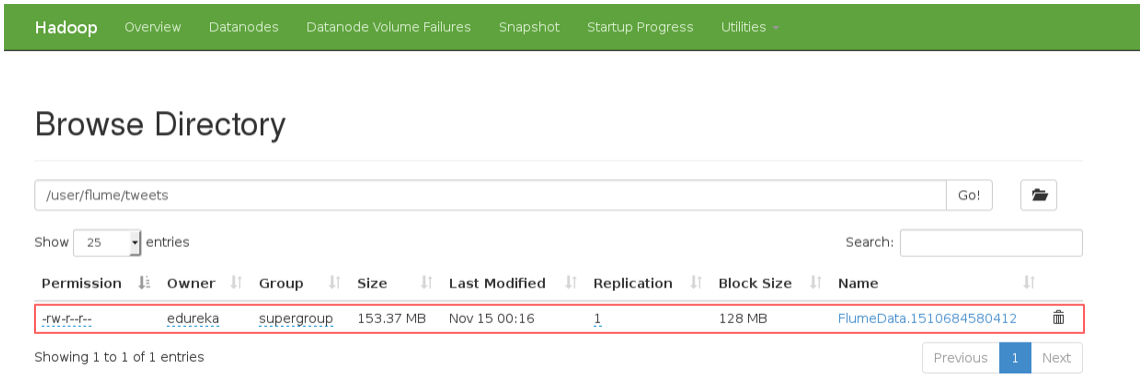
Ahora estamos listos para la ejecución. Sigamos adelante y ejecutemos este comando:

```
$FLUME_HOME/bin/flume-ng agent --conf ./conf/ -f $FLUME_HOME/flume.conf
```

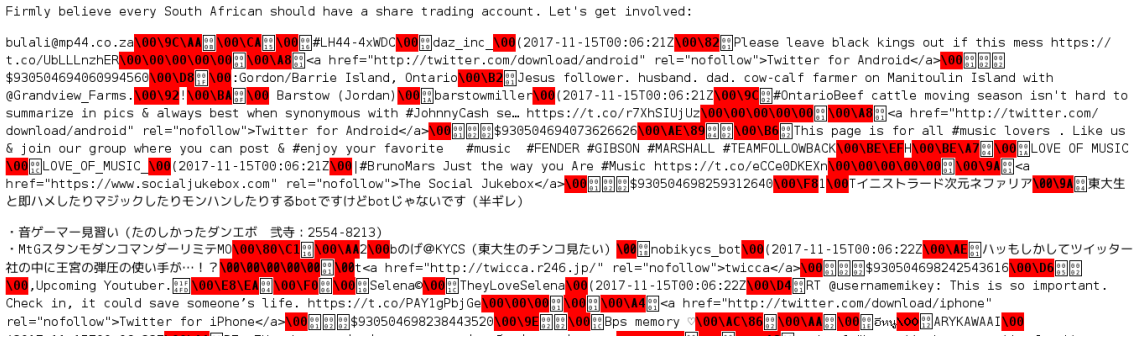
```
[edureka@localhost apache-flume-1.7.0-bin]$ bin/flume-ng agent --conf ./conf/ -f
 flume.conf Dflume.root.logger=DEBUG,console -n TwitterAgent
info: Sourcing environment configuration script /usr/lib/apache-flume-1.7.0-bin/
conf/flume-env.sh
Info: Including Hadoop libraries found via (/usr/lib/hadoop-2.8.1/bin/hadoop) fo
r HDFS access
Info: Including Hive libraries found via () for Hive access
+ exec /usr/lib/jvm/jdk1.8.0_144/bin/java -Xmx20m -cp '/usr/lib/apache-flume-1.7
.0-bin/conf:/usr/lib/apache-flume-1.7.0-bin/lib/*:/usr/lib/hadoop-2.8.1/etc/hado
op:/usr/lib/hadoop-2.8.1/share/hadoop/common/lib/*:/usr/lib/hadoop-2.8.1/share/h
adoop/common/*:/usr/lib/hadoop-2.8.1/share/hadoop/hdfs:/usr/lib/hadoop-2.8.1/sha
re/hadoop/hdfs/lib/*:/usr/lib/hadoop-2.8.1/share/hadoop/hdfs/*:/usr/lib/hadoop-2
.8.1/share/hadoop/yarn/lib/*:/usr/lib/hadoop-2.8.1/share/hadoop/yarn/*:/usr/lib/
hadoop-2.8.1/share/hadoop/mapreduce/lib/*:/usr/lib/hadoop-2.8.1/share/hadoop/map
reduce/*:/usr/lib/hadoop-2.8.1/contrib/capacity-scheduler/*.jar:/lib/*' -Djava.l
ibrary.path=:/usr/lib/hadoop-2.8.1/lib/native org.apache.flume.node.Application
-f flume.conf Dflume.root.logger=DEBUG,console -n TwitterAgent
```

Después de ejecutar este comando, puede salir del terminal usando CTRL + C.

Después puede seguir en su directorio de Hadoop y verificar la ruta mencionada, ya sea que el archivo se haya creado o no.



Descarga el archivo y ábrelo. Obtendrá algo como se muestra en la imagen de abajo.



Ahora que ha entendido a Apache Flume, consulte la capacitación de Hadoop.

Ver :

- https://www.cloudsigma.com/realtime-twitter-data-ingestion-using-flume/
- http://www.barriblog.com/2017/10/lo-siempre-quiso-saber-del-api-twitter-nunca-se-atrevio-preguntar-actualizado-2017/?lang=en