

# Session 12:

## Oozie and Flume

### Assignment 1

- Prachi Mohite

Apache Flume is a Hadoop ecosystem component used to collect, aggregate and moves a large amount of log data from different sources to a centralized data store. It is an open source component which is designed to locate and store the data in a distributed environment and collects the data as per the specified input key(s).

Flume is composed of the following components. **Flume Event:** It is the main unit of the data that is transported inside the **Flume** (Typically a single log entry). It contains a payload of the byte array that is to be transported from the source path to the destination path which could be accompanied by optional

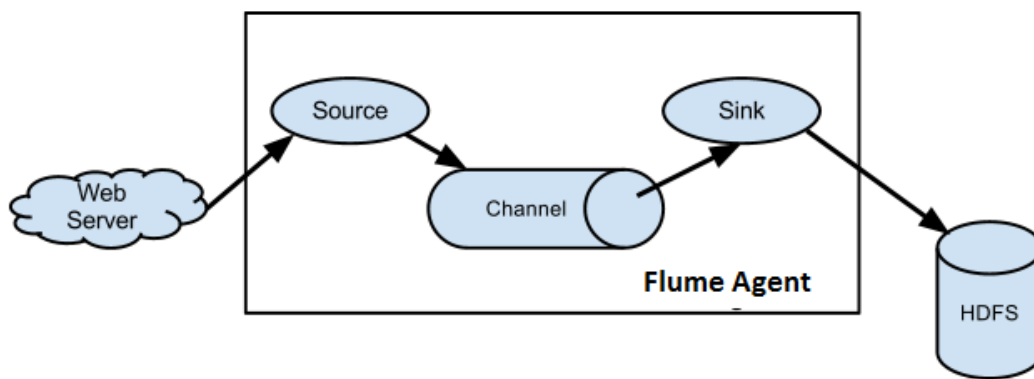
headers.

A Flume

event will be in the following

Header → Payload

**Flume Agent:** Is an independent Java virtual machine daemon process which receives the data (events) from clients and transports to the subsequent destination (sink or agent).



**Source:** Is the component of Flume agent which receives data from the data generators say, twitter, facebook, weblogs from different sites and transfers this data to one or more channels in the form of Flume event. The external source sends data to Flume in a format that is recognized by the target Flume source. Example, an Avro Flume source can be used to receive Avro data from Avro clients or other Flume agents in the flow that send data from an Avro sink, or the Thrift Flume source will receive data from a Thrift sink, or a Flume Thrift RPC client or Thrift Clients are written in any language generated from the Flume thrift protocol.

**Channel:** Once, the Flume source receives an Event, it stores this data into one or more channel and buffers them till they are consumed by sinks. It acts as a bridge between the source and sinks. These channels are implemented to handle any number of sources and sinks.

### Task :

Create a flume agent that streams data from Twitter and stores in the HDFS.

**Streaming Twitter Data** To stream data to our database from twitter we should have the following pre-requisites.

- Twitter account
- Hadoop cluster

If both prerequisites are available we can move to our further step. **Step 1:** Login to the twitter account

1. Login to Twitter Account
2. Go to the following link and click the 'create new app' button. <https://apps.twitter.com/app>
3. Complete all the necessary steps required and 'Create new Twitter Application'

The screenshot shows the Twitter Application Management interface. At the top, there's a header 'Application Management' with a user profile icon. Below the header, a green notification bar states: 'Your application has been created. Please take a moment to review and adjust your application's settings.' The main heading is 'PrachiAcadGildApp' with a 'Test OAuth' button. Below this are tabs for 'Details', 'Settings', 'Keys and Access Tokens', and 'Permissions'. The 'Details' tab is active, showing a Twitter logo icon, the application name 'PrachiAcadGildApp', the description 'Streaming Data from Twitter to HDFS', and the website 'https://www.acadgild.com'. The 'Organization' section is below, with fields for 'Organization' (set to 'None') and 'Organization website' (set to 'None'). The 'Application Settings' section follows, with a note: 'Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.' It contains a table of settings:

Access level	Read and write (modify app permissions)
Consumer Key (API Key)	0SfS80rjdZZlgjrqapR1x6ul (manage keys and access tokens)
Callback URL	None
Callback URL Locked	Yes
Sign in with Twitter	Yes
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

4. Select the 'Keys and Access Token' tab.

Copy the consumer key and the consumer secret code.

Scroll down further and select the 'create my access token' button.

## PrachiAcadGildApp

Test OAuth

DetailsSettingsKeys and Access TokensPermissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	0Sfs80rjdZZlgjjrqapR1x6ul
Consumer Secret (API Secret)	geilMmtEILCEPQKO3oRBhMxL6EPqlpgymTl4dMJedrw070yo3tL
Access Level	Read and write (modify app permissions)
Owner	ExpPrachi
Owner ID	995501284315705345

#### Application Actions

Regenerate Consumer Key and SecretChange App Permissions

#### Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

#### Token Actions

Create my access token

### Status

Your application access token **has been successfully generated**. It may take a moment for changes you've made to reflect.  
[Refresh](#) if your changes are not yet indicated.

## PrachiAcadGildApp

[Test OAuth](#)[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	0SfS80rjdZZlgjjrqapR1x6ul
Consumer Secret (API Secret)	geIMmtEILCEPQKO3oRBhMxL6EPqlpgymT14dMJedrw070yo3tL
Access Level	Read and write ( <a href="#">modify app permissions</a> )
Owner	ExpPrachi
Owner ID	995501284315705345

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	995501284315705345- J6WIANEEgf3Oj4LVFaA93bR4LmEvlG4
Access Token Secret	mbBVYDRYXbhgG2PabAZpmXaT1rqo6Po5rqLi7d6J5j6uS
Access Level	Read and write
Owner	ExpPrachi
Owner ID	995501284315705345

◀ ▶

### Token Actions

[Regenerate My Access Token and Token Secret](#)[Revoke Token Access](#)

Edit the above entries in the configuration Files

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

# Describing/Configuring the source
TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.consumerKey=FYm0KeyT75d0H86sNTsMMnJtf
TwitterAgent.sources.Twitter.consumerSecret=8IWj9j4zumLwmo1OvuqPmSmckDRhkWKNtAgtjj60shx4s8Nkbs
TwitterAgent.sources.Twitter.accessToken=995501284315705345-J6WIANEEgf30j4LVFaA93bR4LmEvlg4
TwitterAgent.sources.Twitter.accessTokenSecret=mbBVYDRYxbhgG2PabAZpmXaT1rqo6Po5rqLi7d6J5j6us

TwitterAgent.sources.Twitter.keywords=hadoop, bigdata, mapreduce, mahout, hbase, nosql
# Describing/Configuring the sink

TwitterAgent.sources.Twitter.keywords= hadoop,election,sports, cricket,Big data

TwitterAgent.sinks.HDFS.channel=MemChannel
TwitterAgent.sinks.HDFS.type=hdfs
TwitterAgent.sinks.HDFS.hdfs.path=hdfs://localhost:8020/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=1000

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

Make sure the HDFS file is working

```
[acadgild@localhost ~]$ jps
6320 SecondaryNameNode
6145 DataNode
6485 ResourceManager
6586 NodeManager
15774 Jps
6046 NameNode
```

Make sure the directory mentioned in configuration file is created n HDFS.

```
[acadgild@localhost ~]$ hadoop fs -mkdir /user/flume
18/05/11 22:35:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -mkdir /user/flume/tweets
18/05/11 22:35:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -ls /user/flume
18/05/11 22:35:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x - acadgild supergroup 0 2018-05-11 22:35 /user/flume/tweets
[acadgild@localhost ~]$
```

```
flume-ng agent -conf c -n TwitterAgent -f /home/acadgild/Desktop/Prachi/Flume/ Twitter.Conf
```

```

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
18/05/14 13:20:18 INFO node.PollingPropertiesFileConfigurationProvider: Configuration provider starting
18/05/14 13:20:18 INFO node.PollingPropertiesFileConfigurationProvider: Reloading configuration file:/home/acadgild/Desktop/Prachi/Flume/Twitter/Conf
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Added sinks: HDFS Agent: TwitterAgent
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO conf:FlumeConfiguration: Processing:HDFS
18/05/14 13:20:18 INFO node.AbstractConfigurationProvider: Creating channels
18/05/14 13:20:18 INFO channel.DefaultChannelFactory: Creating instance of channel MemChannel type memory
18/05/14 13:20:18 INFO node.AbstractConfigurationProvider: Created channel MemChannel
18/05/14 13:20:18 INFO source.DefaultSourceFactory: Creating instance of source Twitter, type org.apache.flume.source.twitter.TwitterSource
18/05/14 13:20:18 INFO sink.DefaultSinkFactory: Creating instance of sink: HDFS, type: hdfs
18/05/14 13:20:18 INFO node.AbstractConfigurationProvider: Channel MemChannel connected to [Twitter, HDFS]
18/05/14 13:20:18 INFO runner.DefaultRunner: Starting new source runners: { source=org.apache.flume.source.twitter.TwitterSource(name=Twitter,state=IDLE) } sinkRun
ners={HDFS=SinkRunner: { policy=org.apache.flume.sink.DefaultSinkProcessor@1e62b4a counter=org.name=null contexts: } } channels={MemChannel=org.apache.flume.channel.MemoryChannel(name=MemChannel)} }
18/05/14 13:20:19 INFO node.Application: Starting Channel MemChannel
18/05/14 13:20:19 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: CHANNEL, name: MemChannel: Successfully registered new MBean.
18/05/14 13:20:19 INFO instrumentation.MonitoredCounterGroup: Component type: CHANNEL, name: MemChannel started
18/05/14 13:20:19 INFO node.Application: Starting Sink HDFS
18/05/14 13:20:19 INFO node.Application: Starting Source Twitter
18/05/14 13:20:19 INFO twitter.TwitterSource: Starting twitter source org.apache.flume.source.twitter.TwitterSource(name=Twitter,state=IDLE)
18/05/14 13:20:19 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SINK, name: HDFS: Successfully registered new MBean.
18/05/14 13:20:19 INFO instrumentation.MonitoredCounterGroup: Component type: SINK, name: HDFS started
18/05/14 13:20:19 INFO twitter.TwitterSource: Twitter source Twitter started.
18/05/14 13:20:19 INFO twitter4j.TwitterStreamImpl: Establishing connection.
18/05/14 13:20:22 INFO twitter4j.TwitterStreamImpl: Connection established.
18/05/14 13:20:22 INFO twitter4j.TwitterStreamImpl: Receiving status stream.
18/05/14 13:20:23 INFO HDFS-HDFSOutputStream: Serializer = TEXT, UseLocalFileSystem = false
18/05/14 13:20:25 INFO HDFS.BucketWriter: Creating hdfs://localhost:8020/user/flume/tweets/FlumeData.152628423676.tmp
18/05/14 13:20:25 INFO twitter.TwitterSource: Processed 100 docs
18/05/14 13:20:25 WARN Native2ByteCodeProvider: Unable to load native byte code library for your platform... using builtin-java classes where applicable
18/05/14 13:20:28 INFO twitter.TwitterSource: Processed 200 docs
18/05/14 13:20:31 INFO twitter.TwitterSource: Processed 300 docs
18/05/14 13:20:34 INFO twitter.TwitterSource: Processed 400 docs
18/05/14 13:20:37 INFO twitter.TwitterSource: Processed 500 docs
18/05/14 13:20:40 INFO twitter.TwitterSource: Processed 600 docs
18/05/14 13:20:42 INFO twitter.TwitterSource: Processed 700 docs
18/05/14 13:20:46 INFO twitter.TwitterSource: Processed 800 docs
18/05/14 13:20:49 INFO twitter.TwitterSource: Processed 900 docs
18/05/14 13:20:52 INFO twitter.TwitterSource: Processed 1,000 docs
18/05/14 13:20:53 INFO twitter.TwitterSource: Total docs indexed: 1,000, total skipped docs: 0
18/05/14 13:20:52 INFO twitter.TwitterSource: 30 docs/second
18/05/14 13:20:52 INFO twitter.TwitterSource: Run took 33 seconds and processed:
18/05/14 13:20:52 INFO twitter.TwitterSource: 0.908 MB/sec sent to index
18/05/14 13:20:52 INFO twitter.TwitterSource: 0.268 MB text sent to index
18/05/14 13:20:52 INFO twitter.TwitterSource: There were 0 exceptions ignored:
18/05/14 13:20:56 INFO twitter.TwitterSource: Processed 1,100 docs
18/05/14 13:20:59 INFO twitter.TwitterSource: Processed 1,200 docs
18/05/14 13:21:02 INFO twitter.TwitterSource: Processed 1,300 docs
18/05/14 13:21:06 INFO twitter.TwitterSource: Processed 1,400 docs
18/05/14 13:21:09 INFO twitter.TwitterSource: Processed 1,500 docs
18/05/14 13:21:12 INFO twitter.TwitterSource: Processed 1,600 docs

```



**The comments from Twitter are loaded into HDFS folder**

```
[acadgild@localhost ~]$ hadoop fs -ls /user/flume/tweets
18/05/14 13:59:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup 8885532 2018-05-14 13:30 /user/flume/tweets/FlumeData.1526284223676
-rw-r--r-- 1 acadgild supergroup 3584352 2018-05-14 13:40 /user/flume/tweets/FlumeData.1526284829820
You have new mail in /var/spool/mail/acadgild
```

## Using the cat command to see the output

[illegible]