

Module 8

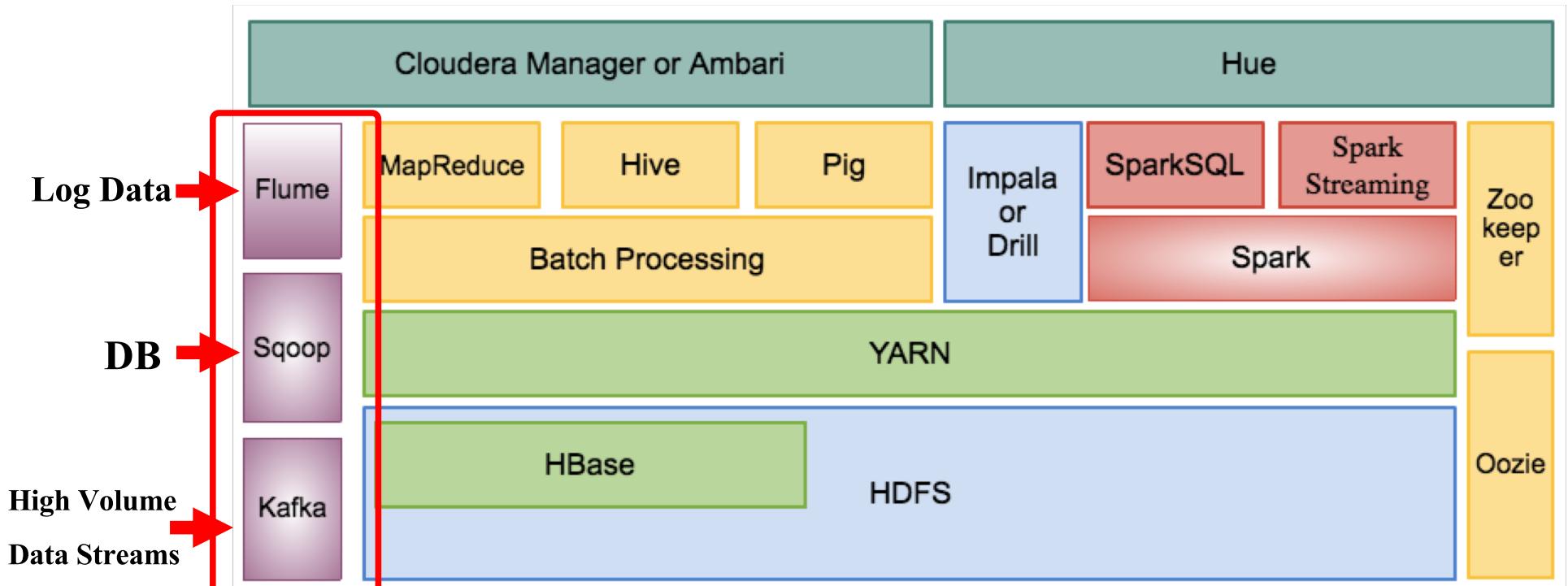
Understanding Flume

Thanachart Numnonda, Executive Director, IMC Institute

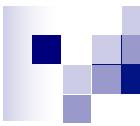
Thanisa Numnonda, Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang

Ingestion

ข้อมูลเข้าใช้ kafka แบบ real time
ข้อมูลน้อยใช้ flume



kafka มีการเก็บข้อมูลให้ด้วย ถ้าข้อมูลเยอะอย่า
ใช้ flume

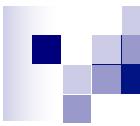


Introduction



Apache Flume is:

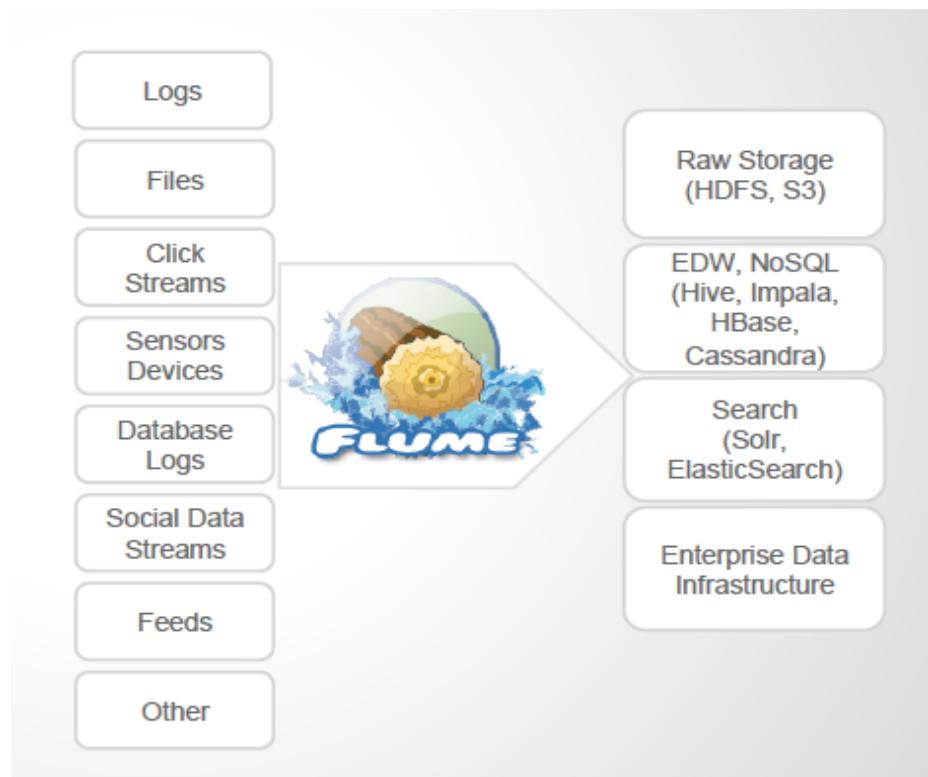
- **A distributed data transport and aggregation system for event- or log-structured data**
- **Principally designed for continuous data ingestion into Hadoop... But more flexible than that**



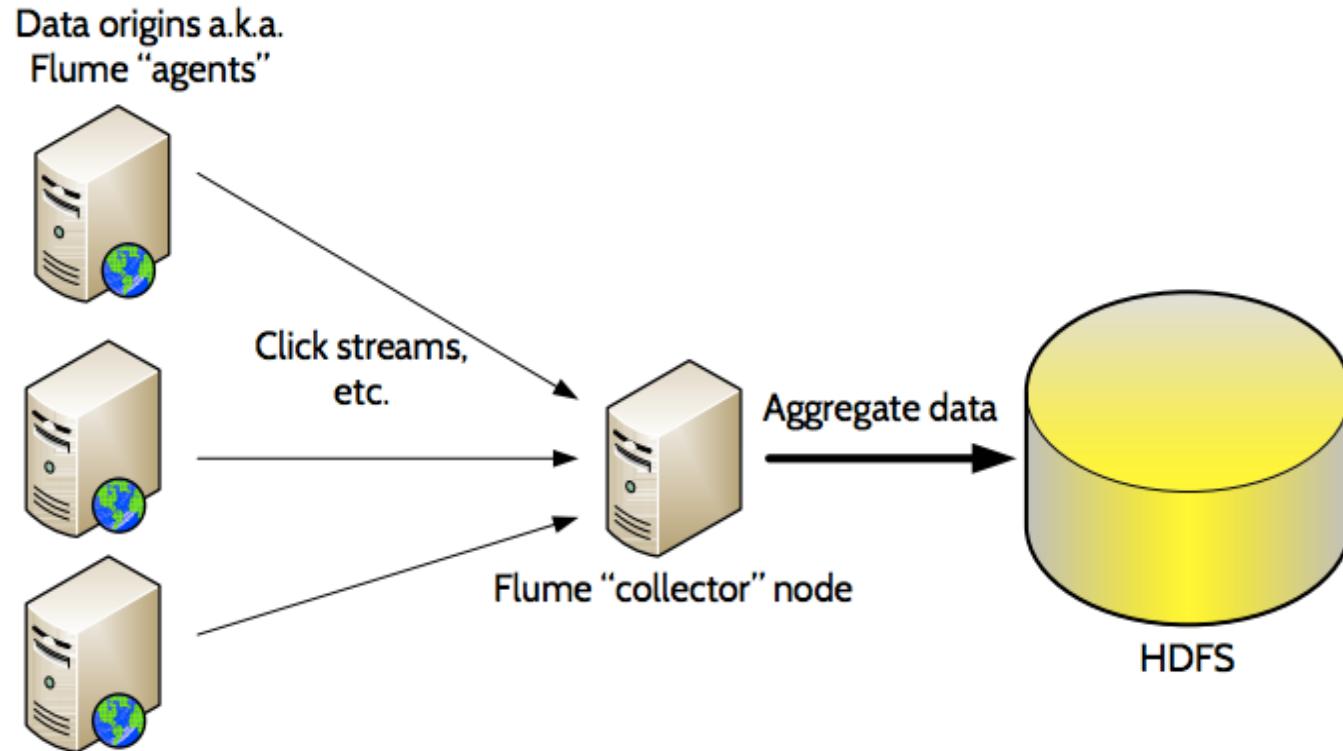
What is Flume?

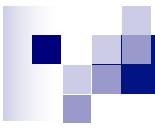
- Apache Flume is a continuous data ingestion system that is...
 - open-source,
 - reliable,
 - scalable,
 - manageable,
 - Customizable,
 - and designed for

Big Data ecosystem

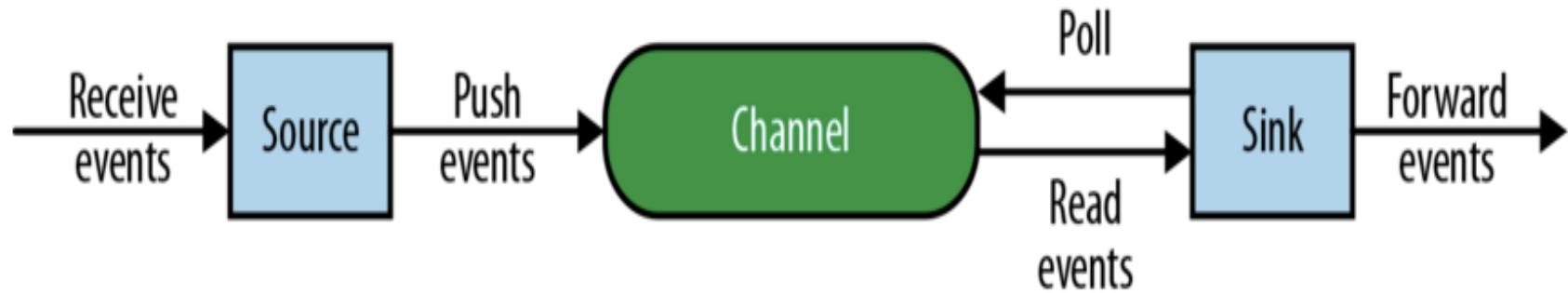


Architecture Overview

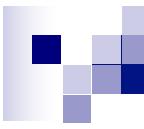




Flume Agent

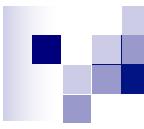


- A source writes events to one or more channels.
- A channel is the holding area as events are passed from a source to a sink.
- A sink receives events from one channel only.
- An agent can have many channels.



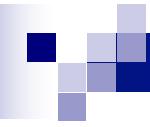
Sources

- Specialized sources for integrating with well-known systems.
 - Example: Spooling Files, Syslog, Netcat, JMS
 - Auto-Generating Sources: Exec, SEQ
 - IPC sources for Agent-to-Agent communication: Avro, Thrift



Channel

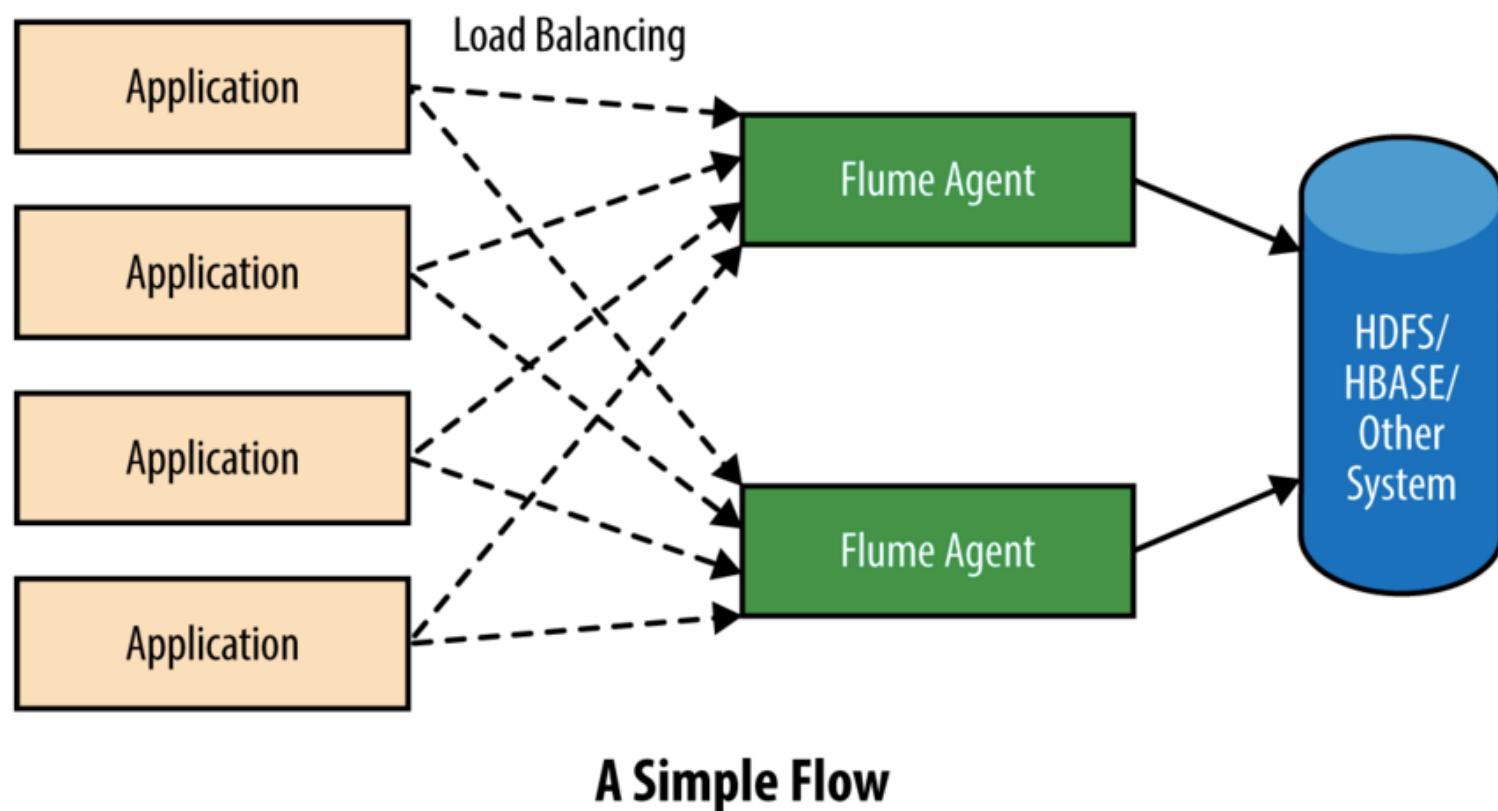
- Different channels offer different levels of persistence:
 - Memory Channel
 - File Channel
 - KafKa Channel
- Channels are fully transactional.
- Provide weak ordering guarantees
- Can work with any number of Sources and Sinks



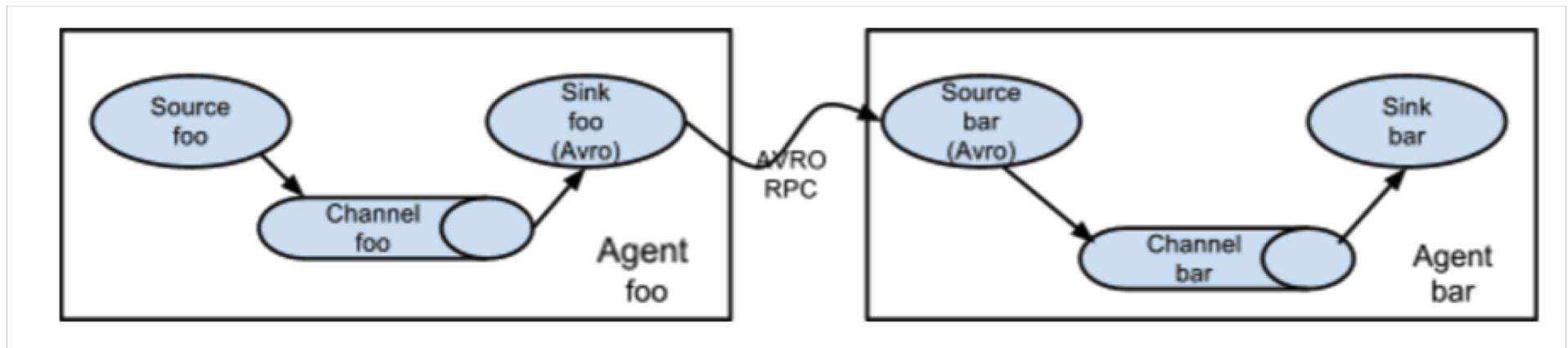
Sink

- Different types of Sinks:
 - Terminal sinks that deposit events to their final destination. For example: HDFS, HBase, Morphline-Solr, Elastic Search, Logger, KafKa
 - Sinks support serialization to user's preferred formats.
 - HDFS sink supports time-based and arbitrary bucketing of data while writing to HDFS.
 - IPC sink for Agent-to-Agent communication: Avro, Thrift
- Require exactly one channel to function

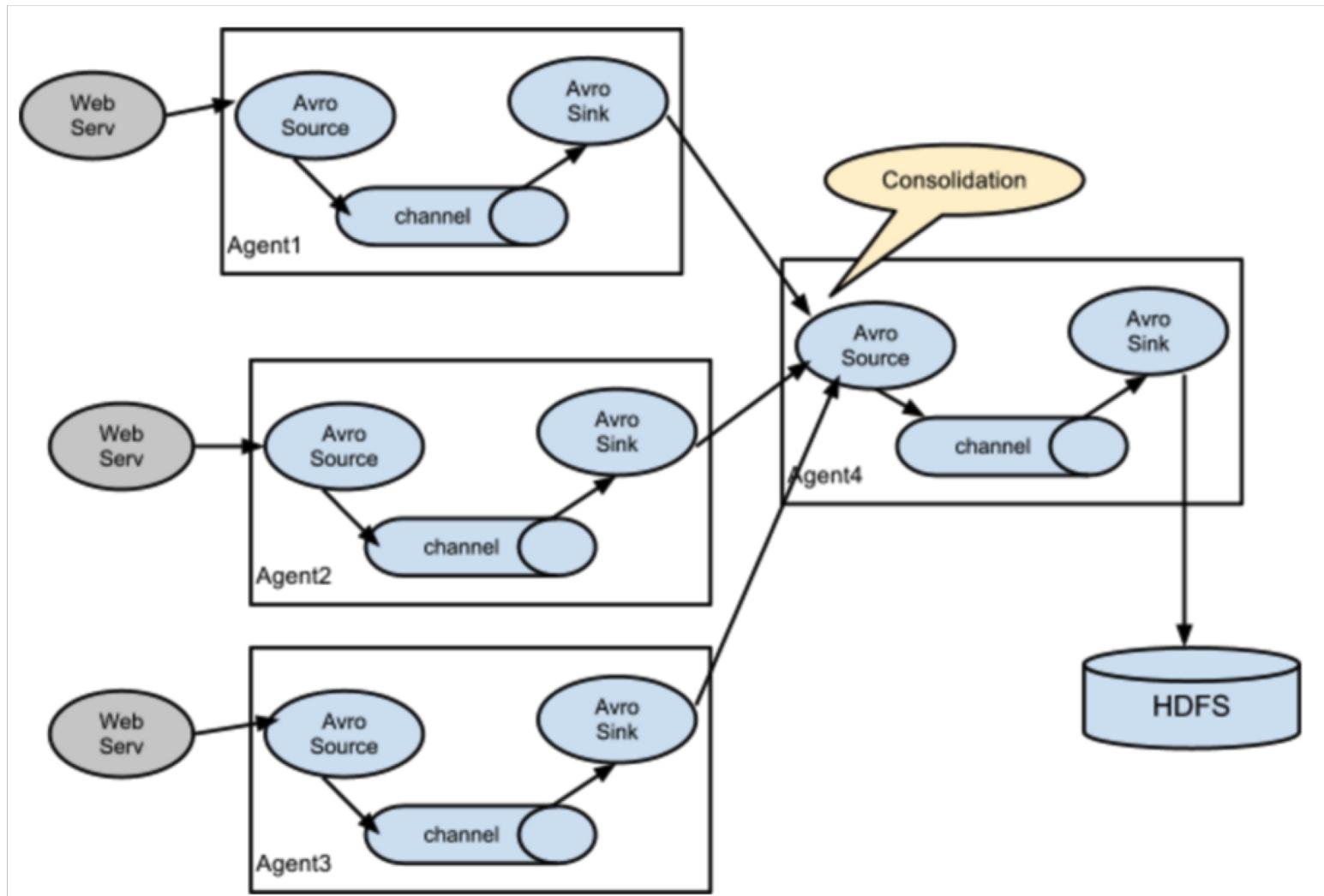
Flow



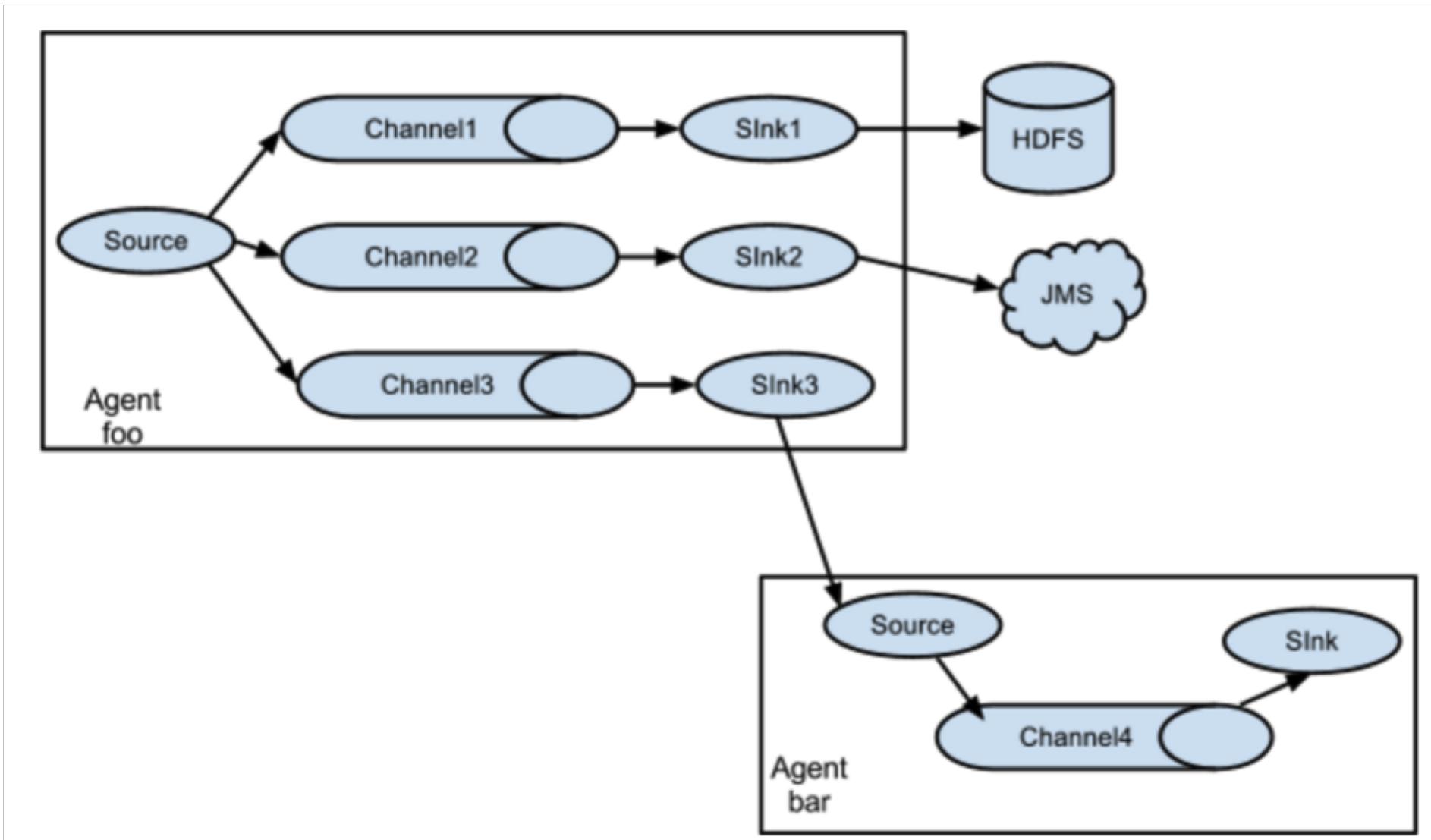
Multi-agent flow

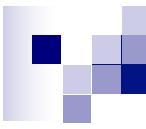


Consolidation



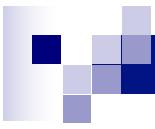
Multiplexing the flow





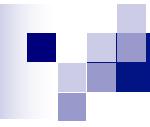
Hands-On: Ingest streaming data using Flume

(LAB 1)



Preparing environment

```
# cd  
  
# mkdir flume  
  
# cd flume  
  
# mkdir conf  
  
# cd conf  
  
# wget https://s3.amazonaws.com/imcbucket/data/example.conf  
  
# nano example.conf
```



Agent Configuration

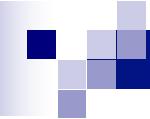
```
a1.sources = r1
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F /opt/gen_logs/logs/access.log

a1.channels = c1

# Use a channel which buffers events to a file
# -- The component type name, needs to be FILE.
a1.channels.c1.type = FILE

# The maximum size of transaction supported by the channel
a1.channels.c1.capacity = 20000
a1.channels.c1.transactionCapacity = 1000

# Amount of time (in millis) between checkpoints
a1.channels.c1.checkpointInterval 3000
```



Agent Configuration

```
# Max size (in bytes) of a single log file
a1.channels.c1.maxFileSize = 2146435071

# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.channel = c1
a1.sinks.k1.hdfs.path = /user/cloudera/flume/%y-%m-%d
a1.sinks.k1.hdfs.filePrefix = flume-%y-%m-%d
a1.sinks.k1.hdfs.rollSize = 1048576
a1.sinks.k1.hdfs.rollCount = 100
a1.sinks.k1.hdfs.rollInterval = 120
a1.sinks.k1.hdfs fileType = DataStream
a1.sinks.k1.hdfs.idleTimeout = 10
a1.sinks.k1.hdfs.useLocalTimeStamp = true

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
a1.sinks = k1
```

Running Flume command

Change a permission of Hadoop directory

```
# sudo -u hdfs hadoop fs -chmod 777 /user
```

```
# flume-ng agent -n a1 -c conf -f /root/flume/conf/example.conf
```

```
19/04/04 09:12:39 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
19/04/04 09:12:39 INFO hdfs.BucketWriter: Creating /user/cloudera/flume/19-04-04/flume-19-04-04.1554369159329.tmp
19/04/04 09:12:42 INFO file.EventQueueBackingStoreFile: Start checkpoint for /root/.flume/file-channel/checkpoint/checkpoint, elements to sync = 10
19/04/04 09:12:42 INFO file.EventQueueBackingStoreFile: Updating checkpoint metadata: logWriteOrderID: 1554369153096, queueSize: 0, queueHead: 16
19/04/04 09:12:42 INFO file.Log: Updated checkpoint for file: /root/.flume/file-channel/data/log-2 position: 2923 logWriteOrderID: 1554369153096
19/04/04 09:12:51 INFO hdfs.BucketWriter: Closing idle bucketWriter /user/cloudera/flume/19-04-04/flume-19-04-04.1554369159329.tmp at 1554369171221
19/04/04 09:12:51 INFO hdfs.BucketWriter: Closing /user/cloudera/flume/19-04-04/flume-19-04-04.1554369159329.tmp
19/04/04 09:12:51 INFO hdfs.BucketWriter: Renaming /user/cloudera/flume/19-04-04/flume-19-04-04.1554369159329.tmp to /user/cloudera/flume/19-04-04/flume-19-04-04.1554369159329
19/04/04 09:12:51 INFO hdfs.HDFSEventSink: Writer callback called.
```

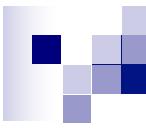


View a result using Hue

Screenshot of the Hue File Browser interface showing a list of files in the directory `/user/cloudera/flume/19-04-04`.

File List:

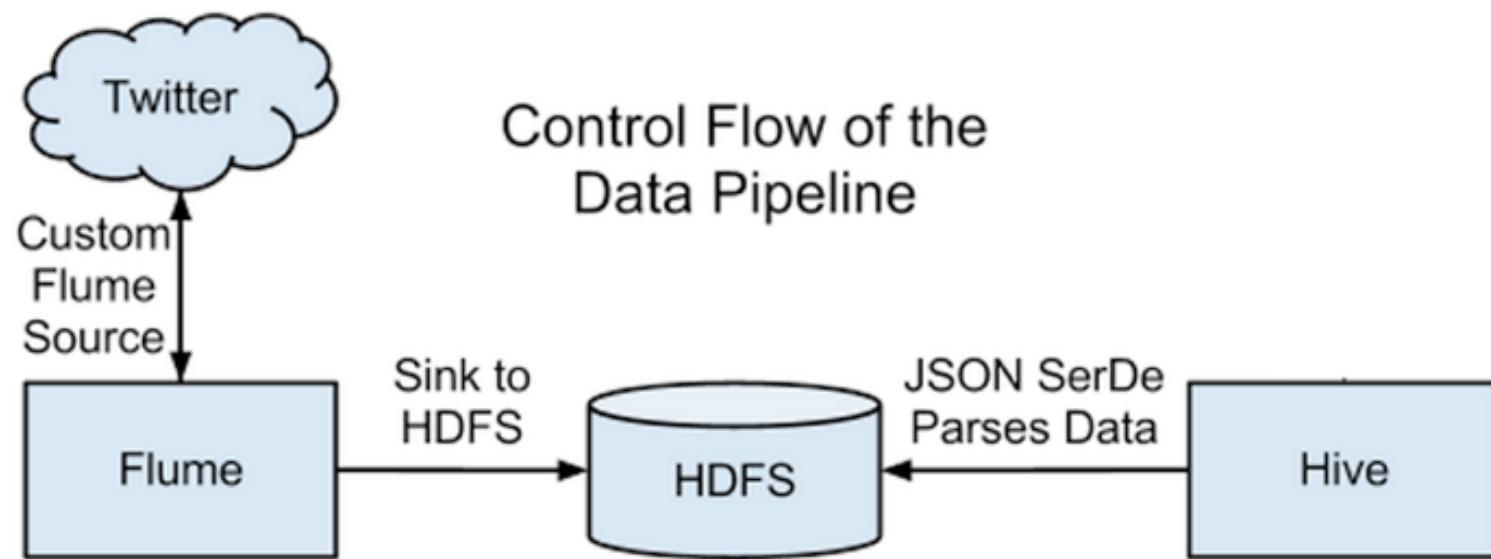
Name	Size	User	Group	Permissions	Date
<code>flume-19-04-04.1554368937856</code>	1.9 KB	root	cloudera	<code>drwxr-xr-x</code>	April 04, 2019 02:08 AM
<code>flume-19-04-04.1554369159329</code>	1.9 KB	root	cloudera	<code>-rw-r--r--</code>	April 04, 2019 02:09 AM
<code>.</code>		root	cloudera	<code>drwxr-xr-x</code>	April 04, 2019 02:12 AM



Hands-On: Loading Twitter Data to Hadoop HDFS

(LAB 2)

Exercise Overview

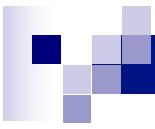


Create a new Twitter App

Login to your Twitter => <https://twitter.com/>

The screenshot shows the Twitter homepage with the following elements:

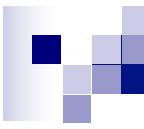
- Header:** Home, Notifications (1), Messages, Twitter logo, Search Twitter, and a blue "Tweet" button.
- User Profile:** A user profile for **thanisa** (@thanisa) with 288 tweets, 32 following, and 121 followers.
- Trends for you:** #ONET (451K Tweets), #เพื่อนกันวันสุดท้าย (4,431 Tweets), and #บุพเพสันนิวาส (796K Tweets).
- What's happening?** A placeholder for a new tweet.
- Tweets:**
 - Pojnsan Ngampongsai** (@Pojnsan_N) · 24h: "c# ไม่น่าจะยากเท่า Java" "jar ยังแมร์ริง ให้แต่โจทย์มา ไม่ให้ hint เชี้ยวเรเลย กูไปไม่เป็นเลย" by น้องนักศึกษาคนหนึ่งบ่น mrt
 - SurveyCompareTH** (@SurveyCompareTH) · 14 Feb 2017: ทำแบบสอบถามแล้วรับเงิน รับ50บาทต่อแบบสอบถาม สมัครฟรีที่นี่ => bit.ly/1UvBznk
- Who to follow:** A sidebar listing users to follow:
 - Hong Kong** (@discoverhk) Follow (Promoted)
 - Followed by K.Chansathit and others
 - tuek** (@phoraphat) Follow
 - NatashaTheRobot** (@Nata...) Follow
- Find people you know:** Import your contacts from Outlook or Connect other address books.



Create a new Twitter App (cont.)

Create a new Twitter App => <https://apps.twitter.com>

The screenshot shows the Twitter Application Management interface. At the top, there is a navigation bar with the Twitter logo and the text "Application Management". On the right side of the bar is a user profile picture and a dropdown arrow. Below the bar, the main title "Twitter Apps" is displayed in large, bold, dark text. Underneath the title, a message box contains the text "You don't currently have any Twitter Apps." A red arrow points from the bottom right towards the word "any". Below the message box is a button labeled "Create New App".



Create a new Twitter App (cont.)

Enter all the details in the application:

Create an application

Application Details

Name *
BD_IT_KMITL_App

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens.

Description *
Demo App @ Big Data Class

Your application description, which will be shown in user-facing authorization screens.

Website *
<http://www.it.kmtl.ac.th>

Your application's publicly accessible home page, where users can go to download, learn about, and give source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Create a new Twitter App (cont.)

Your application will be created:

The screenshot shows the Twitter Application Management interface. At the top, there's a green banner with the message: "Your application has been created. Please take a moment to review and adjust your application's settings." Below this, the application name "BD_IT_KMITL_App" is displayed, along with its logo (a blue Twitter bird icon inside a gear), the organization name "Demo App @ Big Data Class", and the website URL "http://www.it.kmtl.ac.th". There are tabs for "Details", "Settings", "Keys and Access Tokens", and "Permissions", with "Details" currently selected. In the "Application Settings" section, it says "Access level: Read and write (modify app permissions)" and "Consumer Key (API Key): 8kfVSEXgwuqVnINZFiJhXXWKH (manage keys and access tokens)". A "Test OAuth" button is also visible.

Application Management

Your application has been created. Please take a moment to review and adjust your application's settings.

BD_IT_KMITL_App

Demo App @ Big Data Class
http://www.it.kmtl.ac.th

Organization

Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application Settings

Your application's Consumer Key and Secret are used to [authenticate](#) requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
Consumer Key (API Key)	8kfVSEXgwuqVnINZFiJhXXWKH (manage keys and access tokens)

Create a new Twitter App (cont.)

Click on **Keys and Access Tokens**

BD_IT_KMITL_App

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)	8kfvSEXgwuqVnINZFiJhXXWKh
Consumer Secret (API Secret)	BuDIzoKxuleslucGWozw6HvnewZUJ8jhBaT38CPHYDPAvt6Bj
Access Level	Read and write (modify app permissions)
Owner	thanisa
Owner ID	22361688

Create a new Twitter App (cont.)

Your Access token got created:

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.

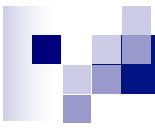
Access Token 22361688-NyJcGTC7qJZ94h8zZEQlVEcGPs4dOmBGDqniP7zsr

Access Token Secret SDt229a5o302y9UKcQRTkthCZMBIWUOMWGoSfrpAT6Be8

Access Level Read and write

Owner thanisa

Owner ID 22361688

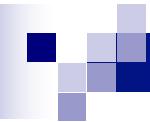


Preparing environment

```
# cd /root/flume/conf
```

```
# wget https://s3.amazonaws.com/imcbucket/data/example2.conf
```

```
# nano example2.conf
```



Agent Configuration

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

TwitterAgent.sources.Twitter.type =
com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey =
<< Your Customer Key >>
TwitterAgent.sources.Twitter.consumerSecret =
<< Your Customer Secret >>
TwitterAgent.sources.Twitter.accessToken =
<< Your Access Token >>
TwitterAgent.sources.Twitter.accessTokenSecret =
<< Your Access Token Secret >>
```

Agent Configuration (cont.)

```
TwitterAgent.sources.Twitter.keywords = hadoop, big data,  
analytics, bigdata, cloudera, data science, data  
scientiest, 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://user/flume/tweets/
```

```
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
```

```
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
```

```
TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000
```

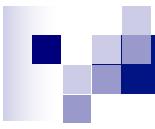
```
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
```

```
TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000
```

```
TwitterAgent.channels.MemChannel.type = memory
```

```
TwitterAgent.channels.MemChannel.capacity = 10000
```

```
TwitterAgent.channels.MemChannel.transactionCapacity = 100
```



Install flume-source

1. **cd** or **mkdir** to

- /usr/lib/flume-ng/lib/
- /usr/lib/flume-ng/plugins.d/twitter-streaming/lib/
- /var/lib/flume-ng/plugins.d/twitter-streaming/lib/

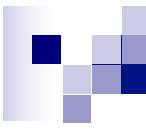
2. **wget** <https://www.dropbox.com/s/lr4u6axfjrrilad/flume-sources-1.0-SNAPSHOT.jar>

Running Flume command and View a result using Hue

```
# flume-ng agent -n TwitterAgent -c conf -f /root/flume/conf/example2.conf
```

The screenshot shows the Hue interface with the 'File Browser' tab selected. The top navigation bar includes links for 'Query Editors', 'Data Browsers', 'Workflows', 'Search', 'File Browser' (which is active), 'Job Browser', 'cloudera', and various help and configuration icons. Below the navigation is a search bar and action buttons for 'Actions', 'Move to trash', 'Upload', and 'New'. The main area displays a file tree under the path '/user/flume/tweets'. A red oval highlights the path '/user/flume/tweets'. The table below lists files with columns for Name, Size, User, Group, Permissions, and Date. All files listed are FlumeData files, mostly from January 25, 2016.

Name	Size	User	Group	Permissions	Date
flume		flume	supergroup	drwxrwxrwx	January 25, 2016 06:06 PM
.		flume	supergroup	drwxrwxrwx	January 25, 2016 06:09 PM
FlumeData.1453773971971	528.0 KB	flume	supergroup	-rw-r--r--	January 25, 2016 06:06 PM
FlumeData.1453774003928	504.7 KB	flume	supergroup	-rw-r--r--	January 25, 2016 06:07 PM
FlumeData.1453774034008	511.9 KB	flume	supergroup	-rw-r--r--	January 25, 2016 06:07 PM
FlumeData.1453774064983	6.8 MB	flume	supergroup	-rw-r--r--	January 25, 2016 06:08 PM
FlumeData.1453774098110	9.9 MB	flume	supergroup	-rw-r--r--	January 25, 2016 06:08 PM
FlumeData.1453774128268	9.9 MB	flume	supergroup	-rw-r--r--	January 25, 2016 06:09 PM
FlumeData.1453774158410.tmp	0 bytes	flume	supergroup	-rw-r--r--	January 25, 2016 06:09 PM



Install Hive json serde

1. **cd** to **/usr/lib/hive/lib/**
2. **wget <http://www.congiu.net/hive-json-serde/1.3.8/cdh5/json-serde-1.3.8-jar-with-dependencies.jar>**

Add jar to Hive

3. Run this command in Hive Editor

```
add jar /usr/lib/hive/lib/json-serde-1.3.8-jar-with-dependencies.jar;
```



Hive Editor Query Editor My Queries Saved Queries History

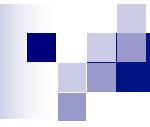
Assist Settings

< default

Tables (0) Q C

The database has no tables

1	add jar /usr/lib/hive/lib/json-serde-1.3.8-jar-with-dependencies.jar
2	
3	
4	
5	



Create Table and Insert Data from HDFS to Hive

4. Run this command to create a table “tweets” and put data created by flume

```
CREATE EXTERNAL TABLE tweets ( id BIGINT, created_at STRING, source STRING,  
favorited BOOLEAN, retweeted_status STRUCT< text:STRING,  
user:STRUCT<screen_name:STRING,name:STRING>, retweet_count:INT>, entities  
STRUCT< urls:ARRAY<STRUCT<expanded_url:STRING>>,  
user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,  
hashtags:ARRAY<STRUCT<text:STRING>>>, text STRING, user STRUCT<  
screen_name:STRING, name:STRING, friends_count:INT, followers_count:INT,  
statuses_count:INT, verified:BOOLEAN, utc_offset:INT, time_zone:STRING>,  
in_reply_to_screen_name STRING ) ROW FORMAT SERDE  
'org.openx.data.jsonserde.JsonSerDe' LOCATION '/user/flume/tweets';
```



Query Table in Hive

5. Run this command to show contents in the table tweets

```
select * from tweets;
```

	Recent queries	Query	Log	Columns	Results	Chart	X	X
	tweets.id	tweets.created_at	tweets.source		tweets.favorited	tweets.retweeted_status		
1	980378951989284864	Sun Apr 01 09:38:51 +0000 2018	SocialTweetHP		false	NULL		
2	980378958255534080	Sun Apr 01 09:38:52 +0000 2018	Twitter for Android		false	{"text":"With ConnectingCare, our first dApp for #HealthN		
3	980378994876051456	Sun Apr 01 09:39:01 +0000 2018	Twitter Web Client		false	NULL		
4	980379012856889345	Sun Apr 01 09:39:05 +0000 2018	Twitter for iPhone		false	{"text":"Antara current marketable job:\n\n1. Engineering		
5	980379014350213120	Sun Apr 01 09:39:06 +0000 2018	SocialTweetHP		false	NULL		
6	980379023401504773	Sun Apr 01 09:39:08 +0000 2018	Twitter for iPad		false	{"text":"Bulding a Unicorn? The funding stages from start		
7	980379041969688577	Sun Apr 01 09:39:12 +0000 2018	Twitter Web Client		false	{"text":"The #MediChain team are #SavingLivesWithBlock		
8	980379051289440256	Sun Apr 01 09:39:15 +0000 2018	Twitter for Android		false	{"text":".@guardian @LeaveEUOfficial @vote_leave @El		
9	980379056603586560	Sun Apr 01 09:39:16 +0000 2018	Paper.li		false	NULL		
10	980379057580924928	Sun Apr 01 09:39:16 +0000 2018	Twitter Web Client		false	{"text":"#MediChain has been using #machinelearning in		
11	980379058440757249	Sun Apr 01 09:39:16 +0000 2018	Twitter Web Client		false	NULL		
12	980379065344495616	Sun Apr 01 09:39:18 +0000 2018	Twitter for iPhone		false	{"text":"Non-#tech businesses begin to use #ArtificialIntell		

Source : https://github.com/madooding/bd_it_kmitl_summarization/blob/master/TwitterMining.md