Module 8

Understanding Flume

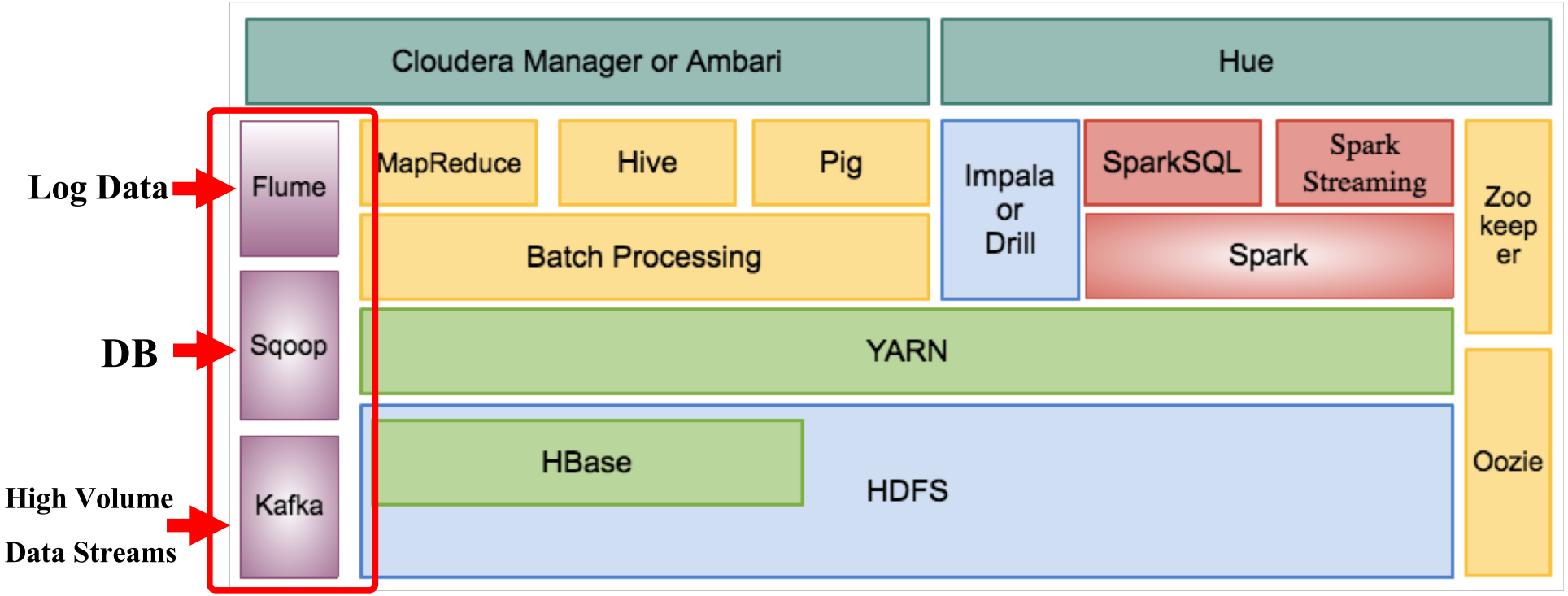
Thanachart Numnonda, Executive Director, IMC Institute

Thanisa Numnonda, Faculty of Information Technology,

King Mongkut's Institute of Technology Ladkrabang

Ingestion

ข้อมูลเยอะใช้ kafka แบบ real time
ข้อมูลน้อยใช้ flume



kafka มีการเก็บข้อมูลให้ด้วย ถ้าข้อมูลเยอะอย่าใช้ flume

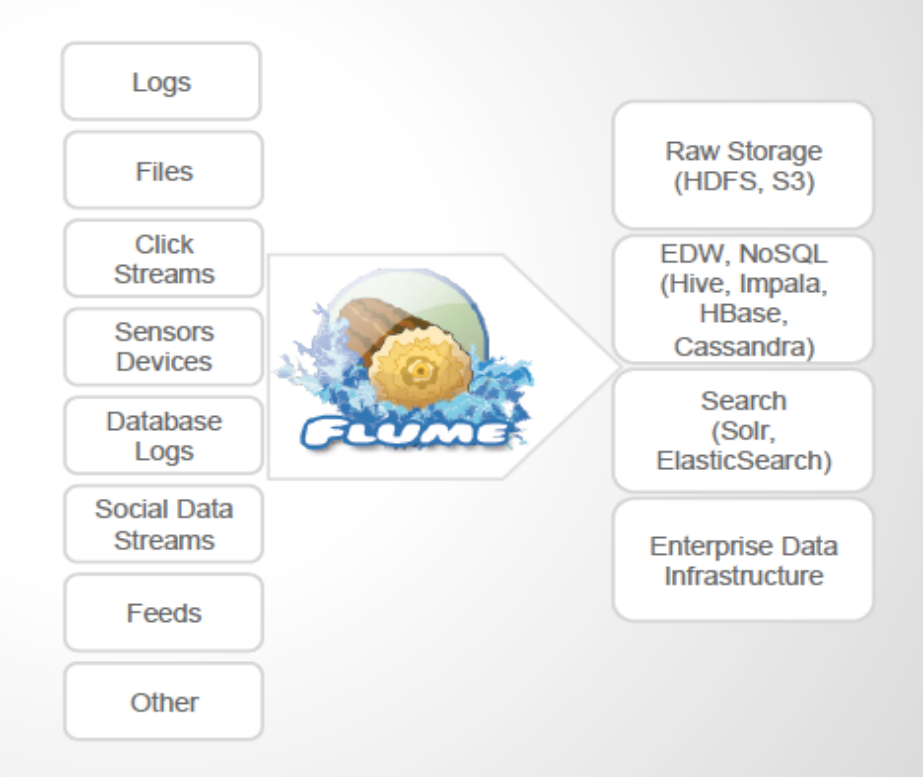
Introduction

Apache Flume is:

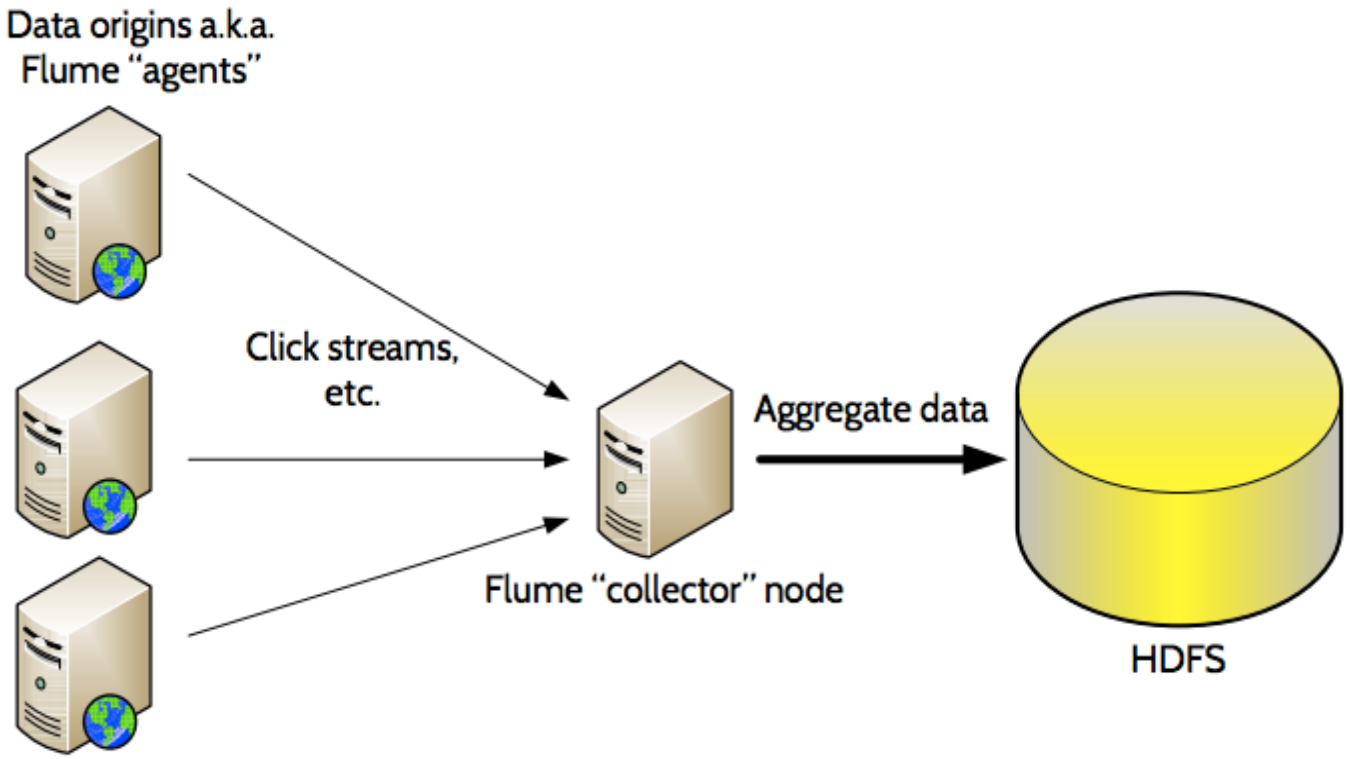
- **A distributed data transport and aggregation system for event- or log-structured data**
- **Principally designed for continuous data ingestion into Hadoop… But more flexible than that**

What is Flume?

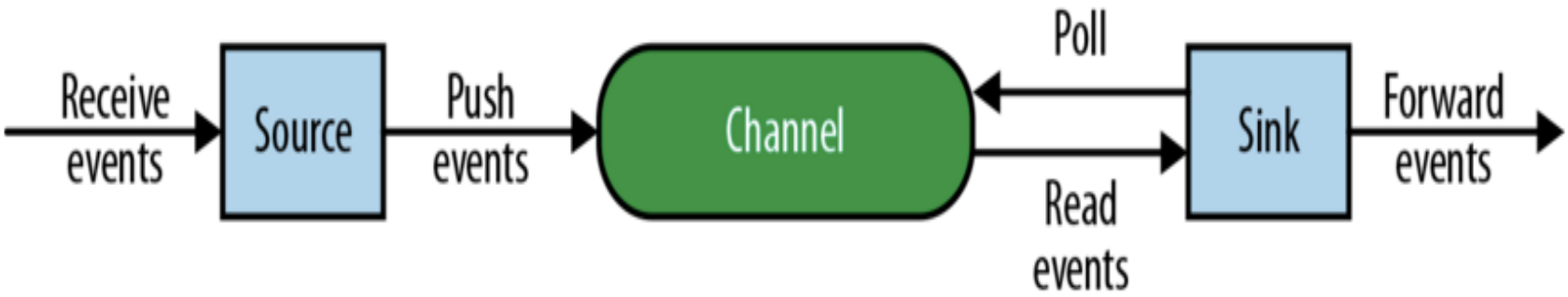
- Apache Flume is a continuous data ingestion system that is...
  - open-source,
  - reliable,
  - scalable,
  - manageable,
  - Customizable,
  - and designed for Big Data ecosystem



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.

Source: Using Flume, Hari Shreedharan, 2014

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

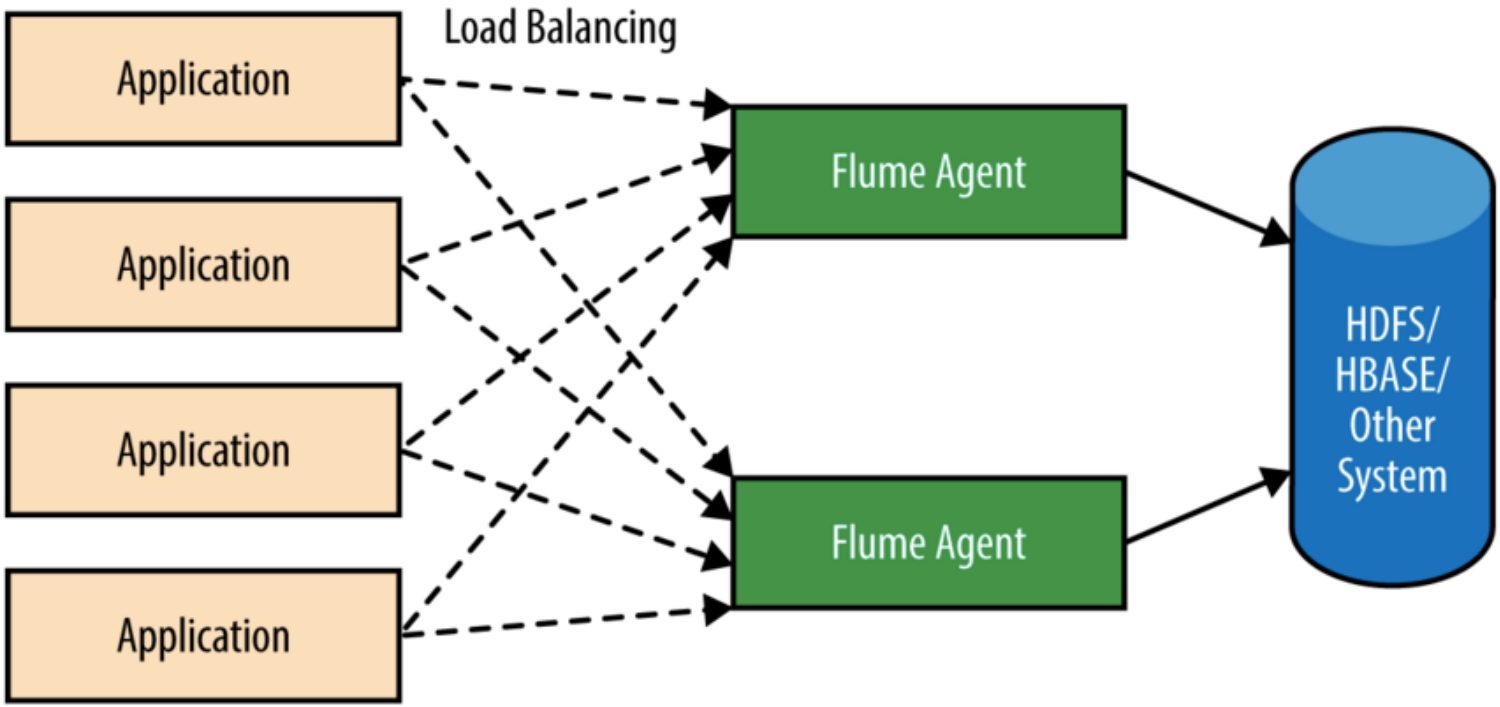
Channel

- 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

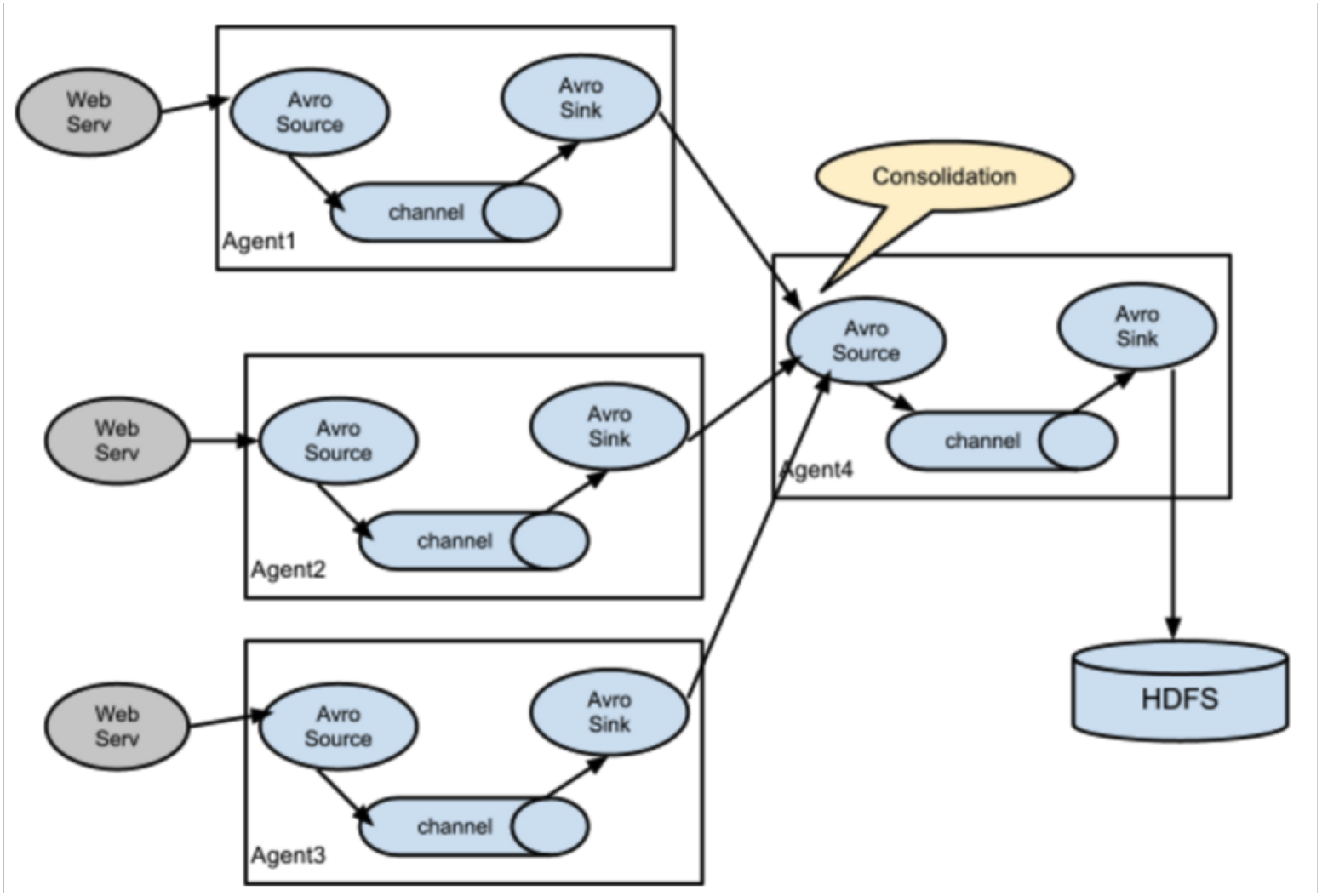


A Simple Flow

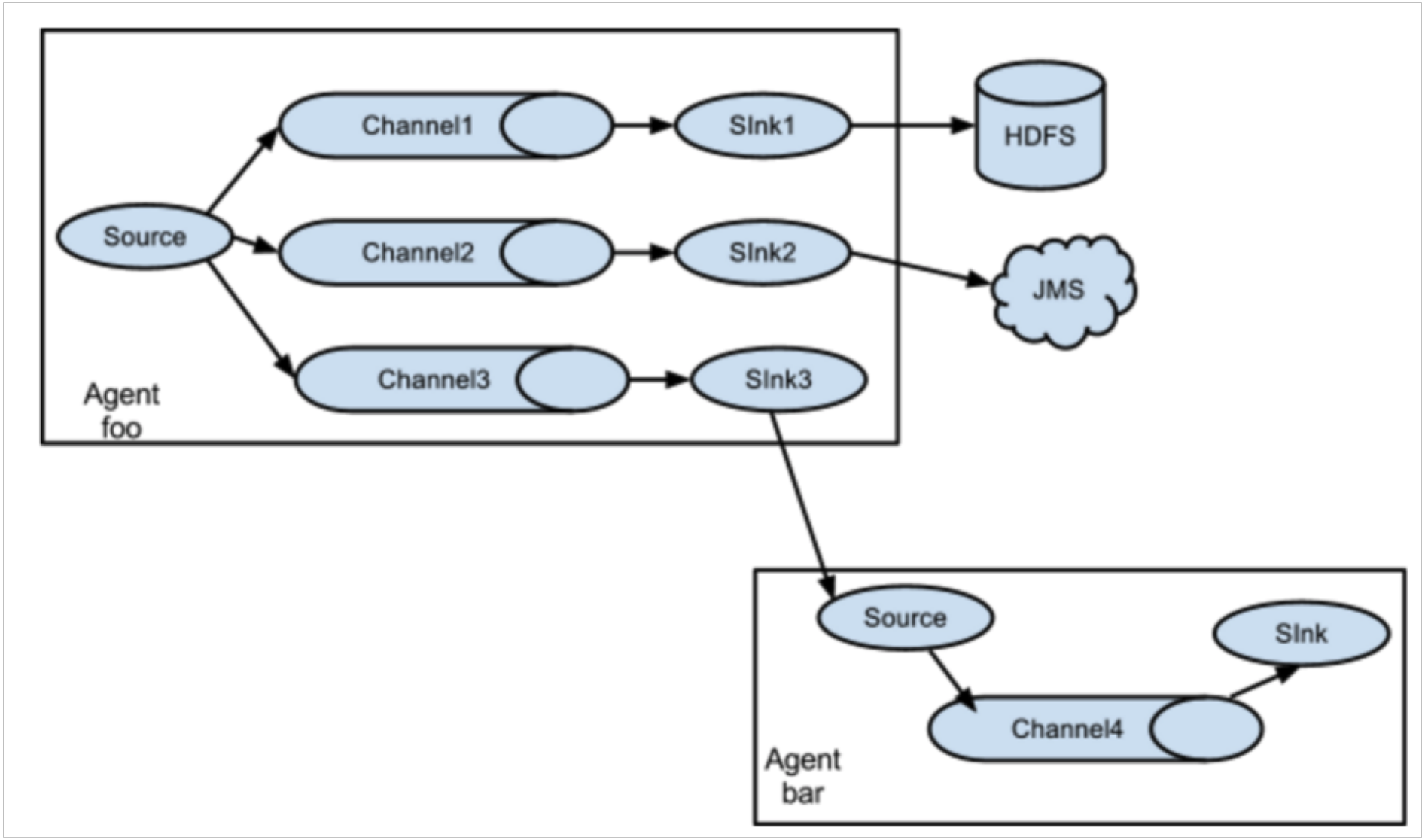
Source: Using Flume, Hari Shreedharan, 2014

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

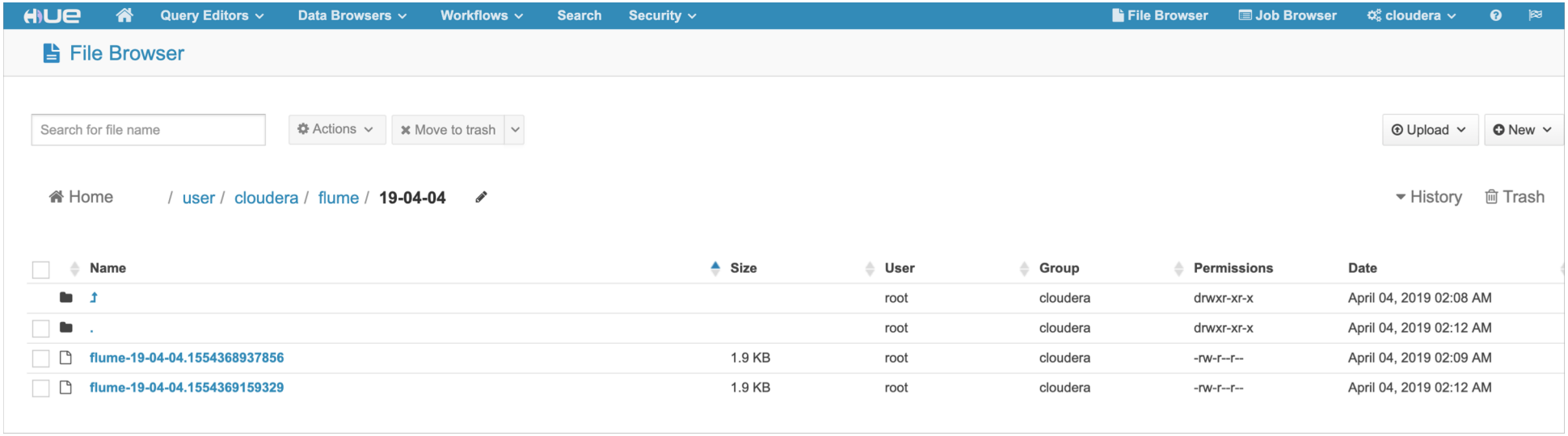
Change a permission of Hadoop directory

```
# 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/fi
le-channel/checkpoint/checkpoint, elements to sync = 10
19/04/04 09:12:42 INFO file.EventQueueBackingStoreFile: Updating checkpoint metadata: logWri
teOrderID: 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.
```

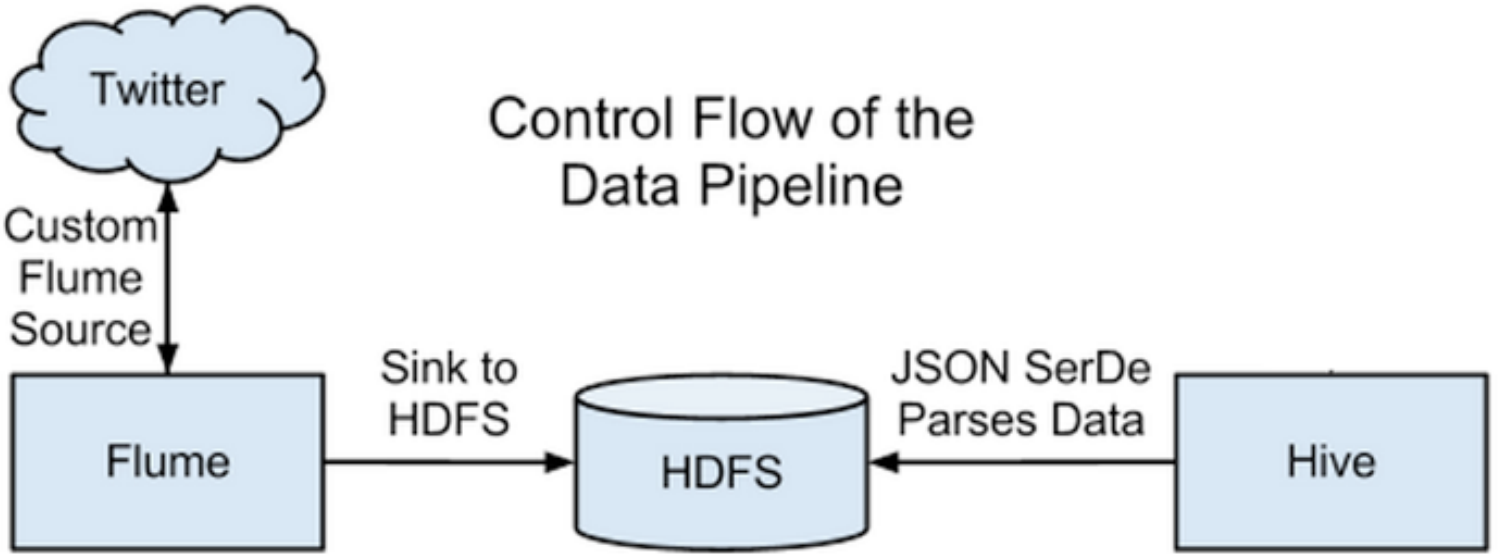
View a result using Hue



Hands-On: Loading Twitter Data to Hadoop HDFS

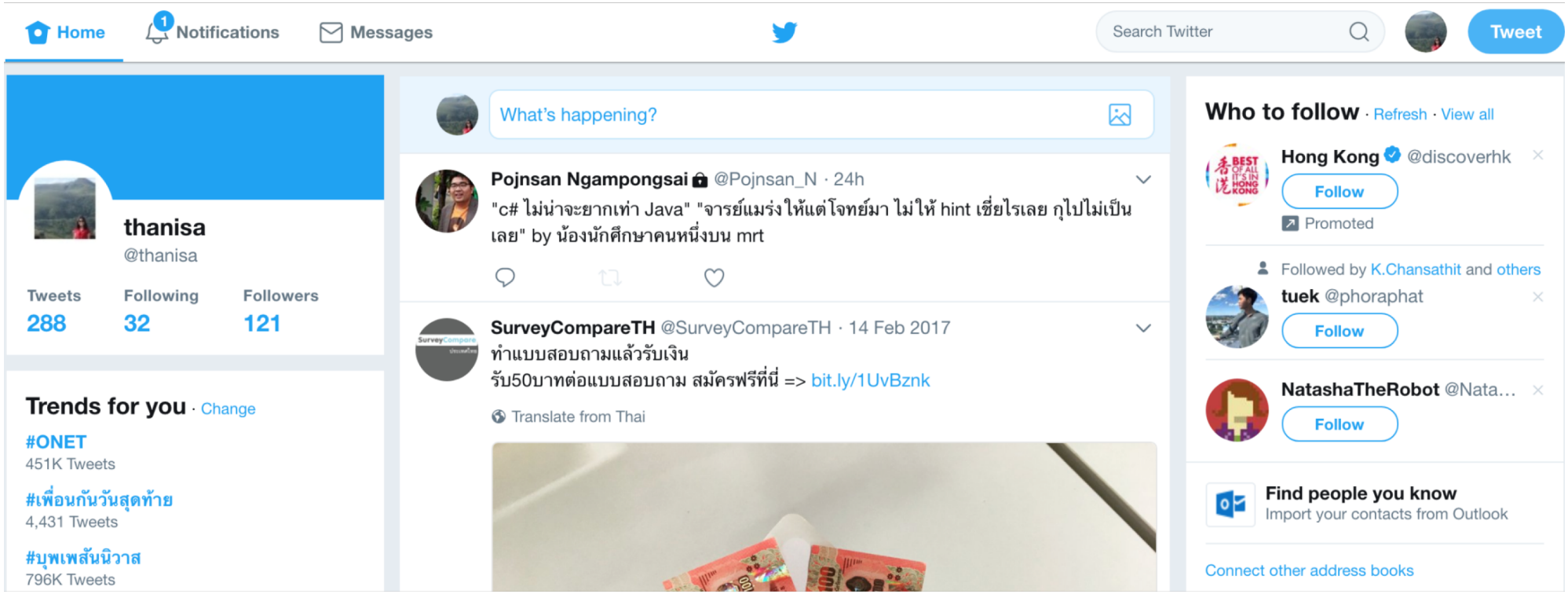
(LAB 2)

Exercise Overview



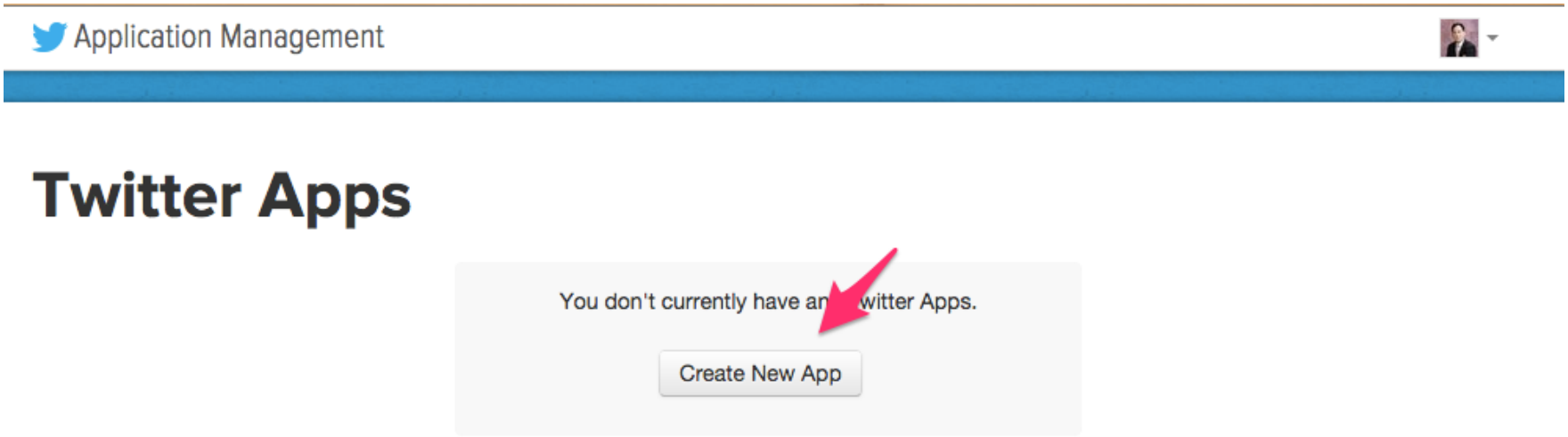
Create a new Twitter App

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



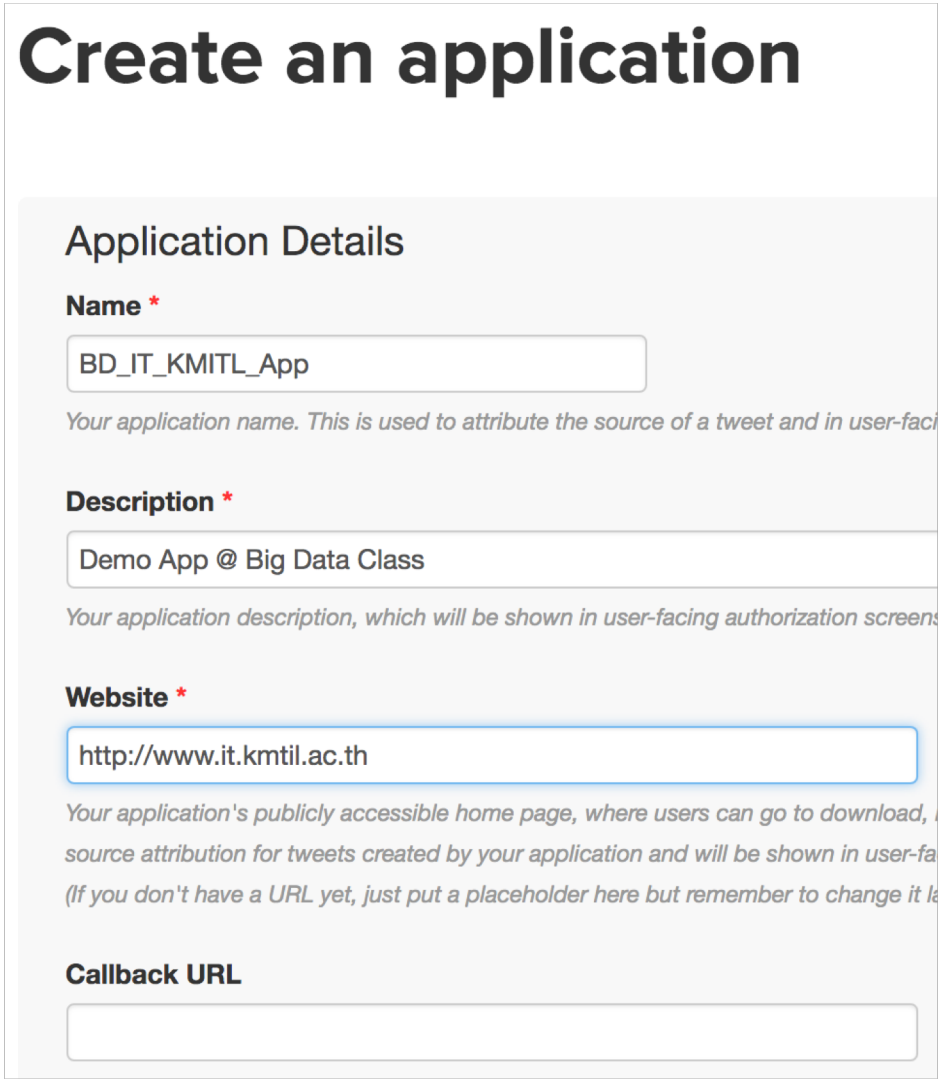
Create a new Twitter App (cont.)

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



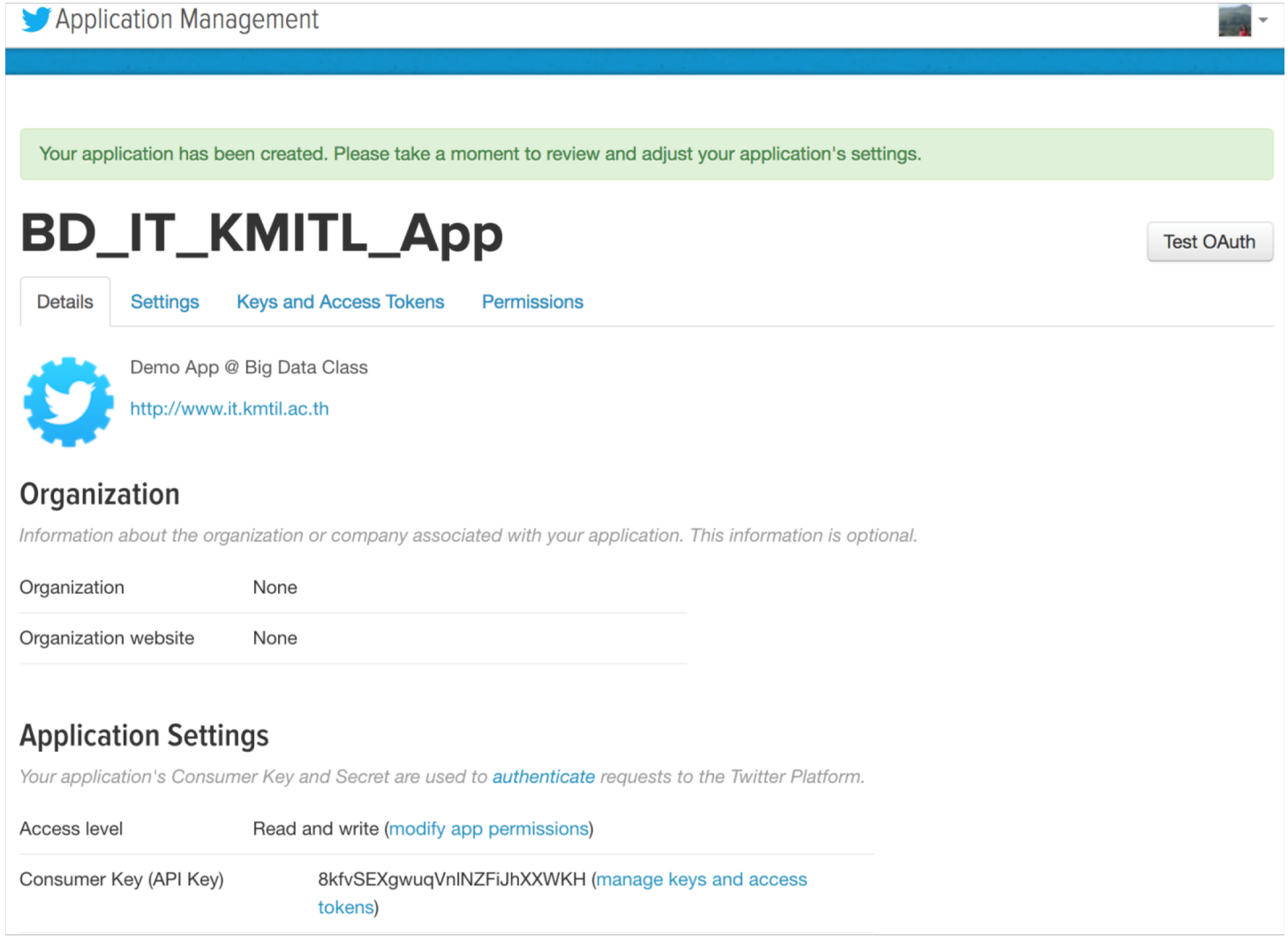
Create a new Twitter App (cont.)

Enter all the details in the application:



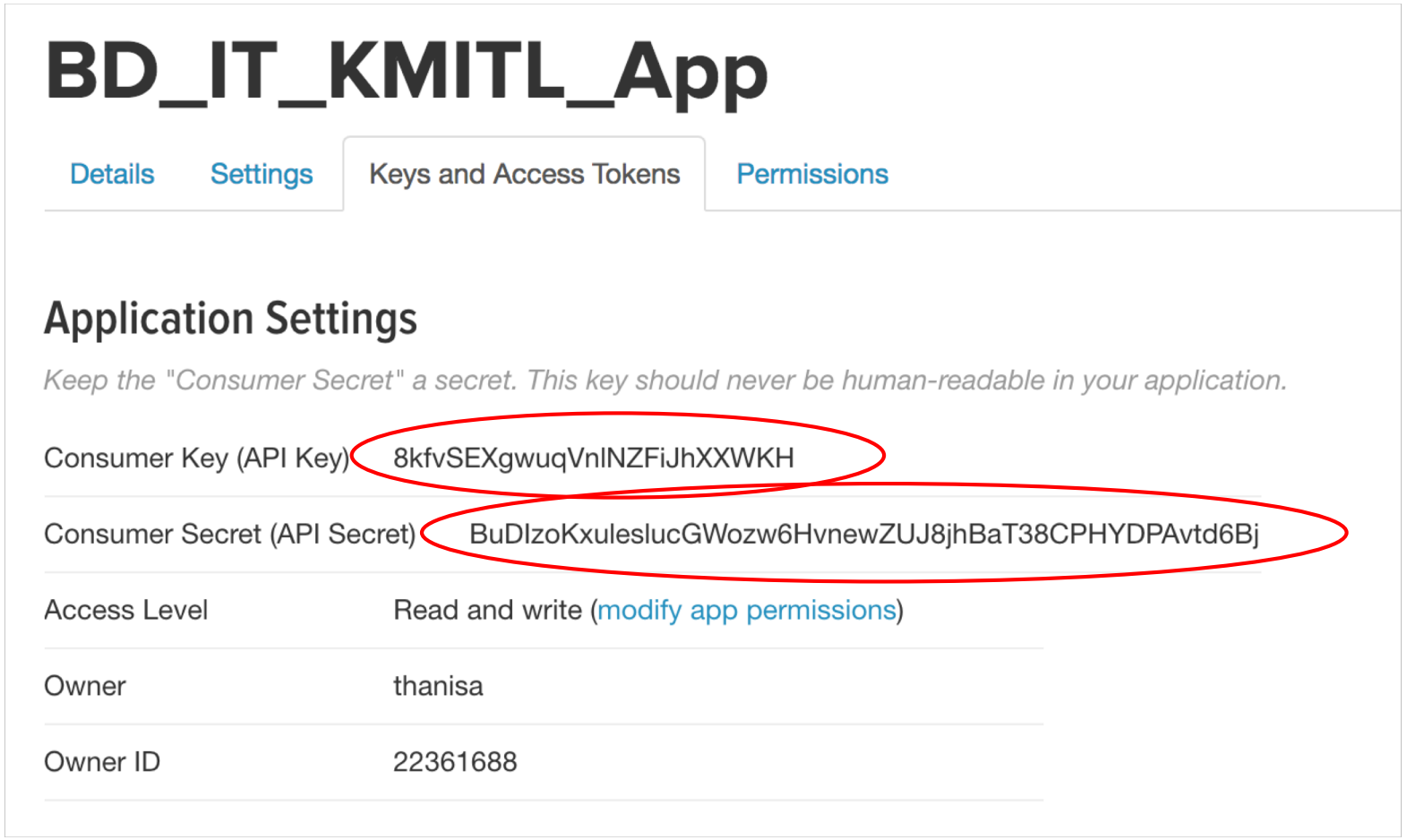
Create a new Twitter App (cont.)

Your application will be created:



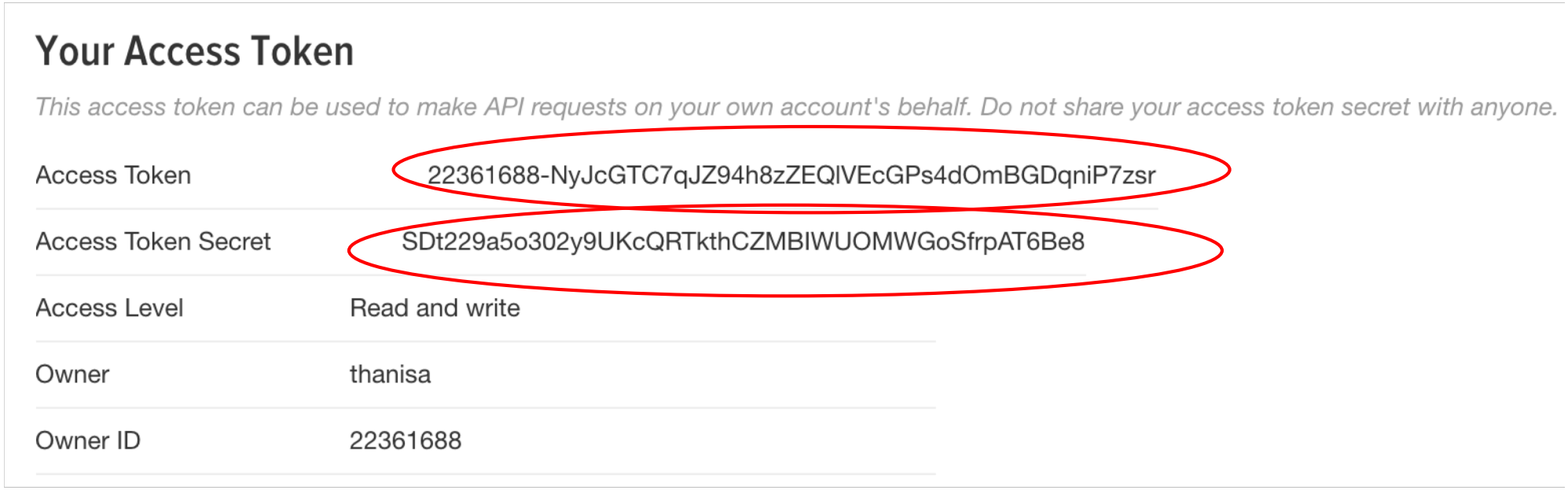
Create a new Twitter App (cont.)

Click on Keys and Access Tokens



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
```

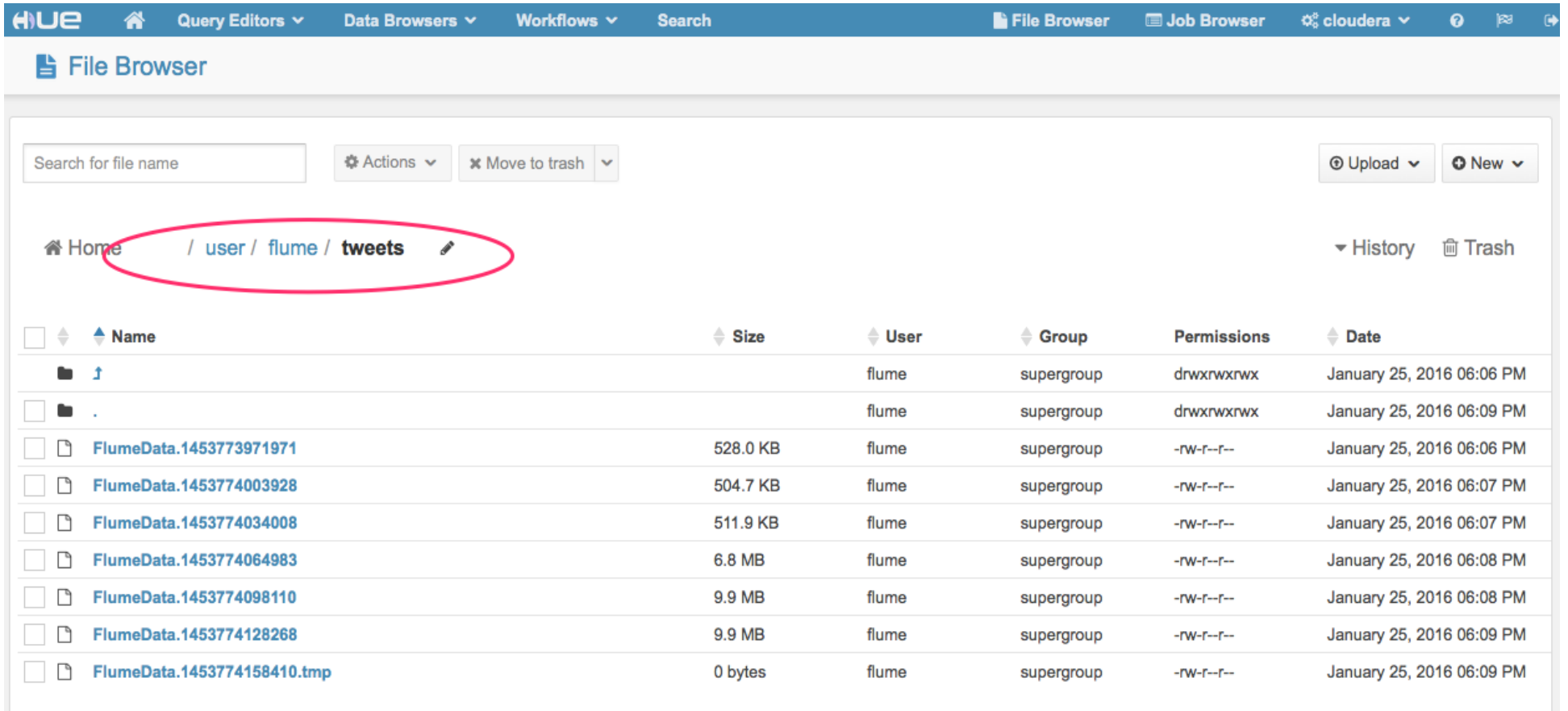
Install flume-source

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
```



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;
```

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

	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 #SavingLivesWithBloc
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 #ArtificialIntel

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