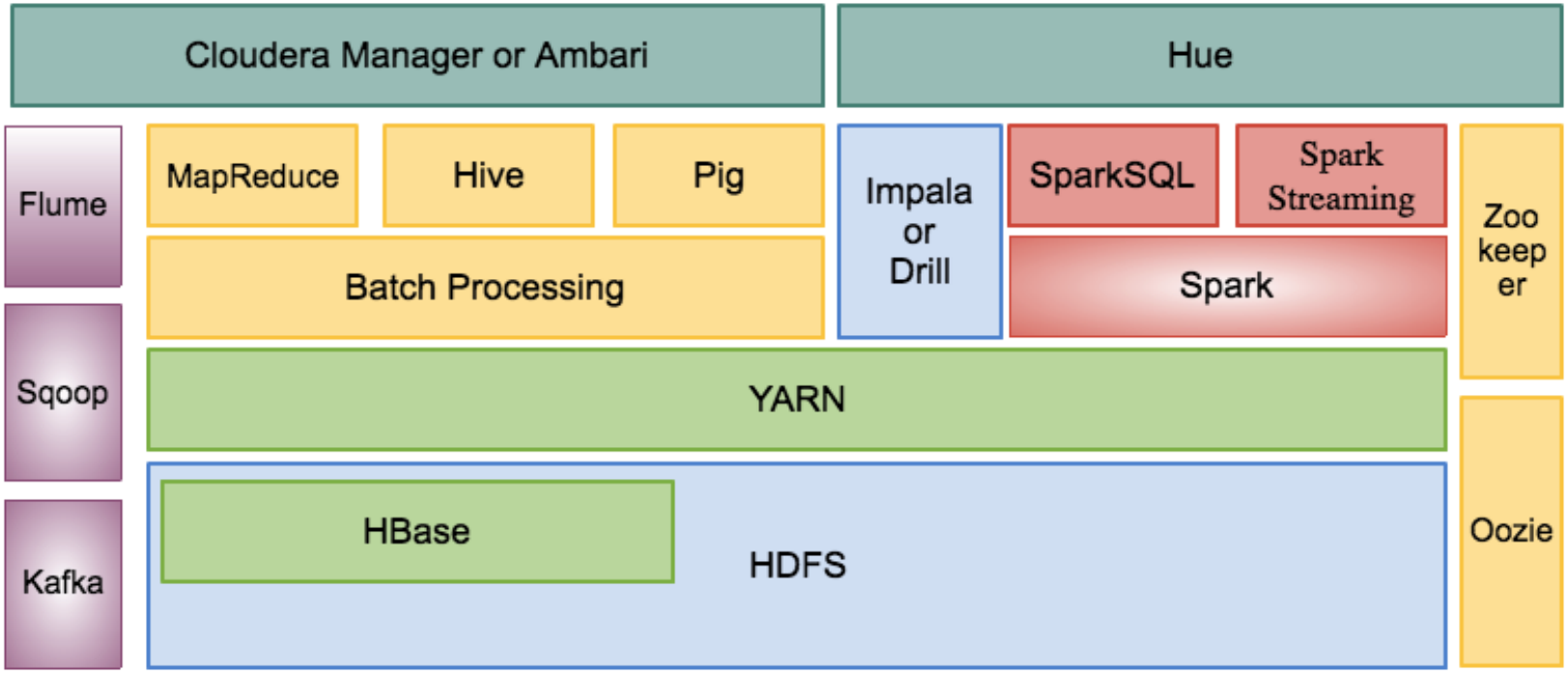
Module 8

Understanding Flume

Thanachart Numnonda, Executive Director, IMC Institute

Thanisa Numnonda, Faculty of Information Technology,

King Mongkut's Institute of Technology Ladkrabang


Cloudera Manager or Ambari
Hue
Flume
MapReduce
Hive
Pig
Impala or Drill
SparkSQL
Spark Streaming
Zoo keep er
Batch Processing
Spark
Sqoop
YARN
Kafka
HBase
HDFS
Oozie

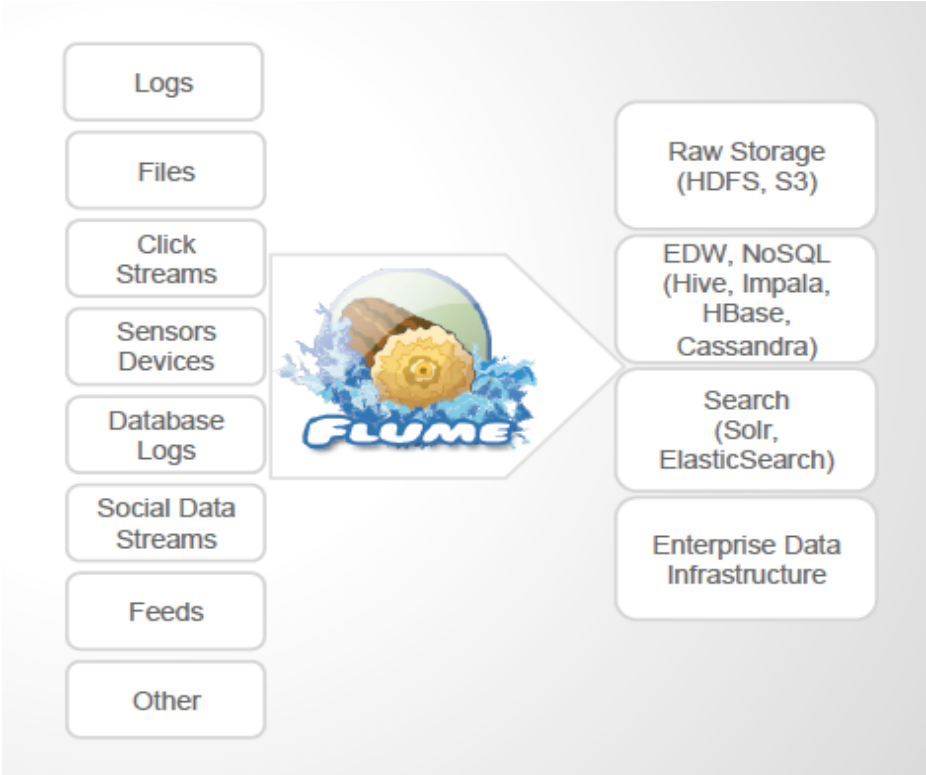
Introduction

Apache Flume is:

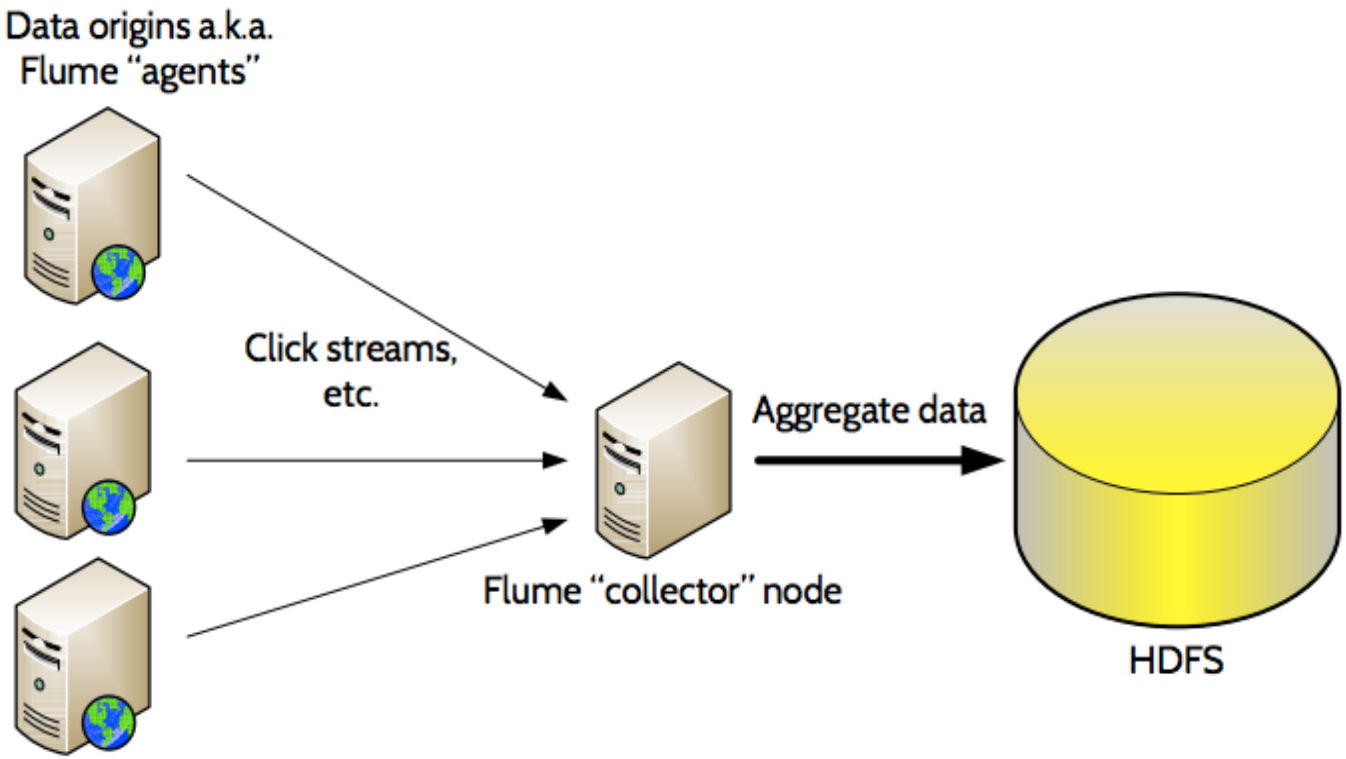
- **A distributed data transport and aggregation system for event- or log-structured data**
- **Principally designed for continuous data ingestion into Hadoop… But more flexible than that**

What is Flume?

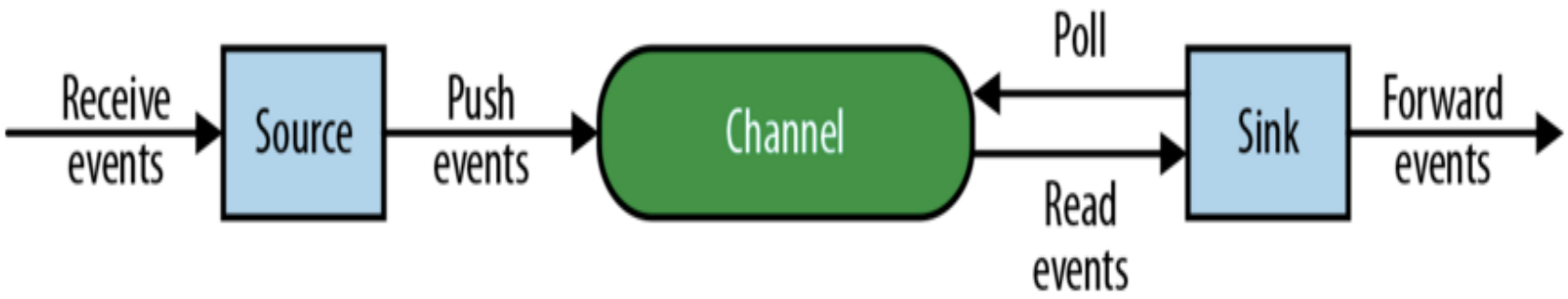
- Apache Flume is a continuous data ingestion system that is...
  - open-source,
  - reliable,
  - scalable,
  - manageable,
  - Customizable,
  - and designed for Big Data ecosystem



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

- Different Source types:
- Require at least one channel to function
- 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:

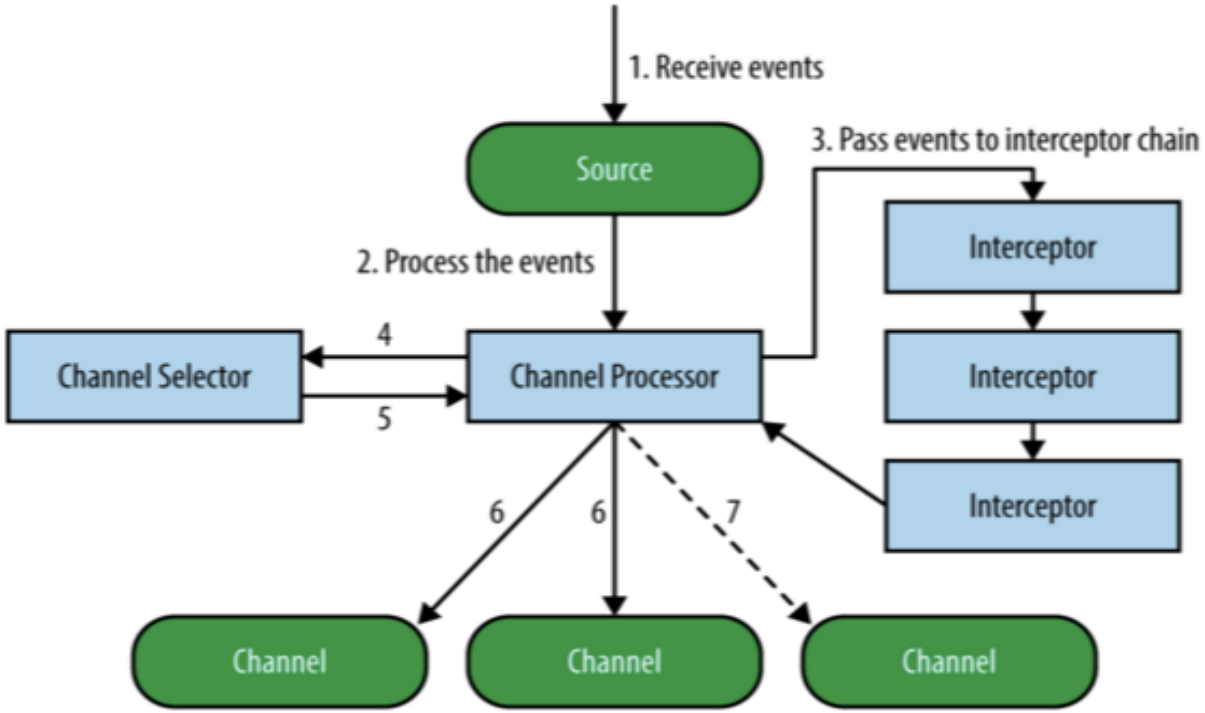
-KafKa Channel:

- Eventually, when the agent comes back data can be accessed.
- 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

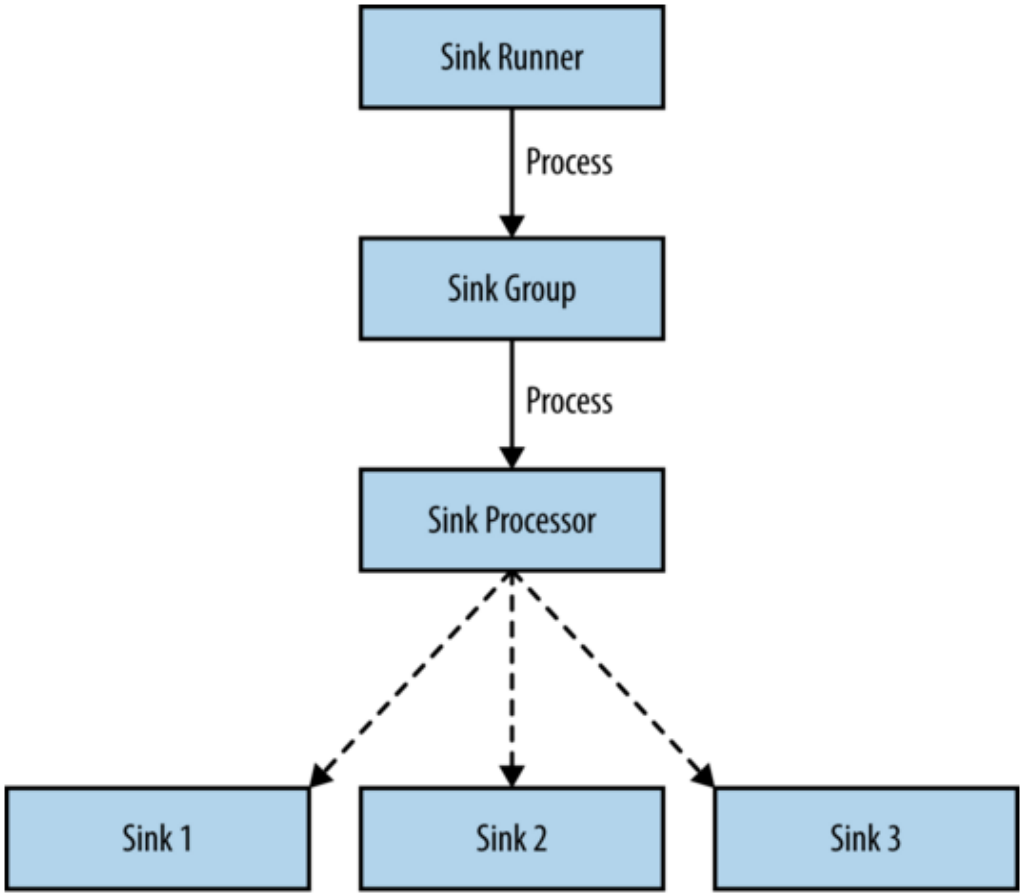
Flume Process



4. Pass each event to channel selector
5. Return list of channels the event is to be written to
6. Write all events that need to go to each required channel. Only one transaction is opened. with each channel, and all events to a channel are written as part of that transaction.
7. Repeat the same with optional channels

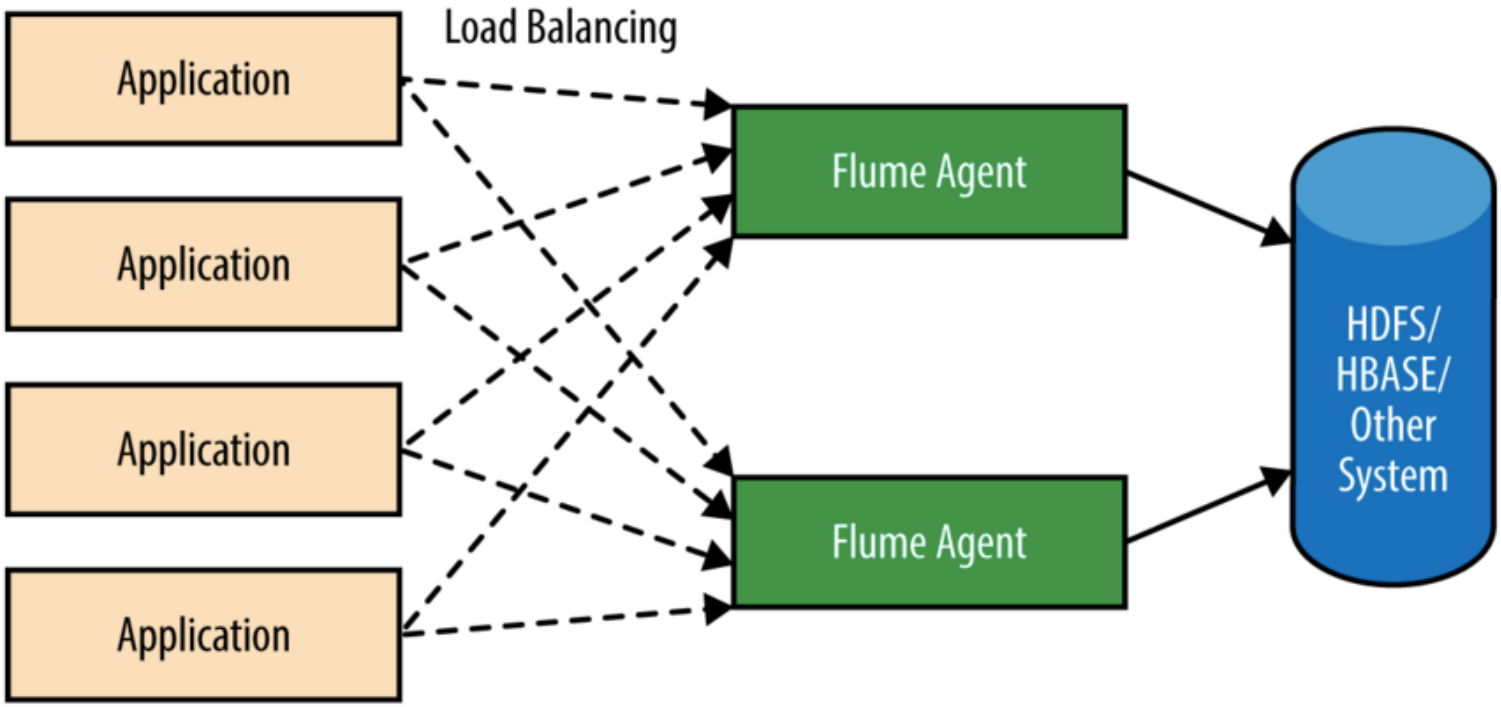
Source: Using Flume, Hari Shreedharan, 2014

Flume Process



Source: Using Flume, Hari Shreedharan, 2014

Flow



A Simple Flow

Source: Using Flume, Hari Shreedharan, 2014

Flume terminology

- 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.

Flume Agent Configuration : Example

```
agent.sources = httpSrc
agent.channels = memory1 memory2
agent.sinks = hdfsSink hbaseSink

agent.sources.httpSrc.type = http
agent.sources.httpSrc.channels = memory1 memory2

# Bind to all interfaces
agent.sources.httpSrc.bind = 0.0.0.0
agent.sources.httpSrc.port = 4353

# Removing this line will disable SSL
agent.sources.httpSrc.ssl = true
agent.sources.httpSrc.keystore = /tmp/keystore
agent.sources.httpSrc.keystore-password = UsingFlume

agent.sources.httpSrc.handler = usingflume.ch03.HTTPSourceXMLHandler
agent.sources.httpSrc.handler.insertTimestamp = true

agent.sources.httpSrc.interceptors = hostInterceptor
agent.sources.httpSrc.interceptors.hostInterceptor.type = host
```

Flume Agent Configuration : Example

```
# Initializes a memory channel with default configuration
agent.channels.memory1.type = memory

# Initializes a memory channel with default configuration
agent.channels.memory2.type = memory

# HDFS Sink
agent.sinks.hdfsSink.type = hdfs
agent.sinks.hdfsSink.channel = memory1
agent.sinks.hdfsSink.hdfs.path = /Data/UsingFlume/%{topic}/%Y/%m/%d/%H/%M
agent.sinks.hdfsSink.hdfs.filePrefix = UsingFlumeData

agent.sinks.hbaseSink.type = asynchbase
agent.sinks.hbaseSink.channel = memory2
agent.sinks.hbaseSink.serializer = usingflume.ch05.AsyncHBaseDirectSerializer
agent.sinks.hbaseSink.table = usingFlumeTable
```

Flume Command

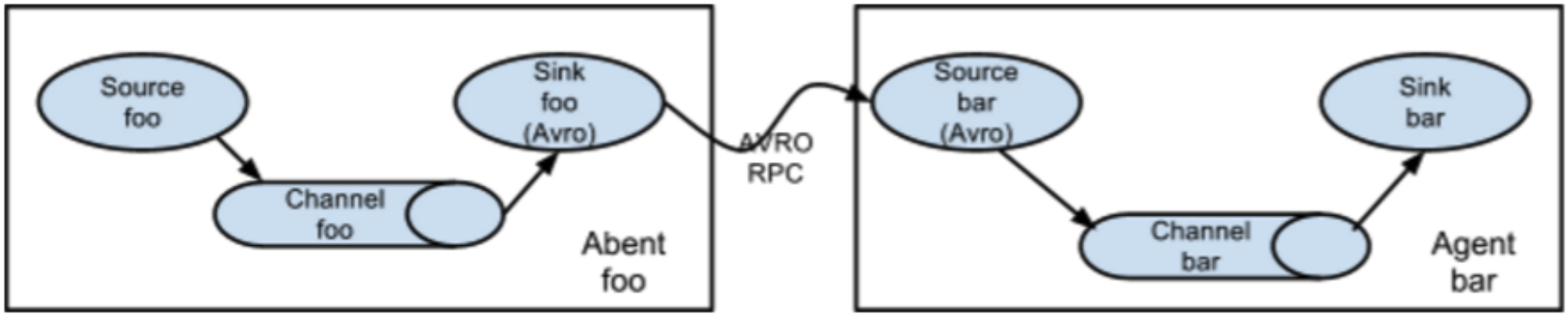
- Check whether flume has installed correctly or not

```
[root@quickstart /]# flume-ng agent -n $agent_name -c
/etc/flume-ng/conf.empty -f conf/flume-
conf.properties.template
```

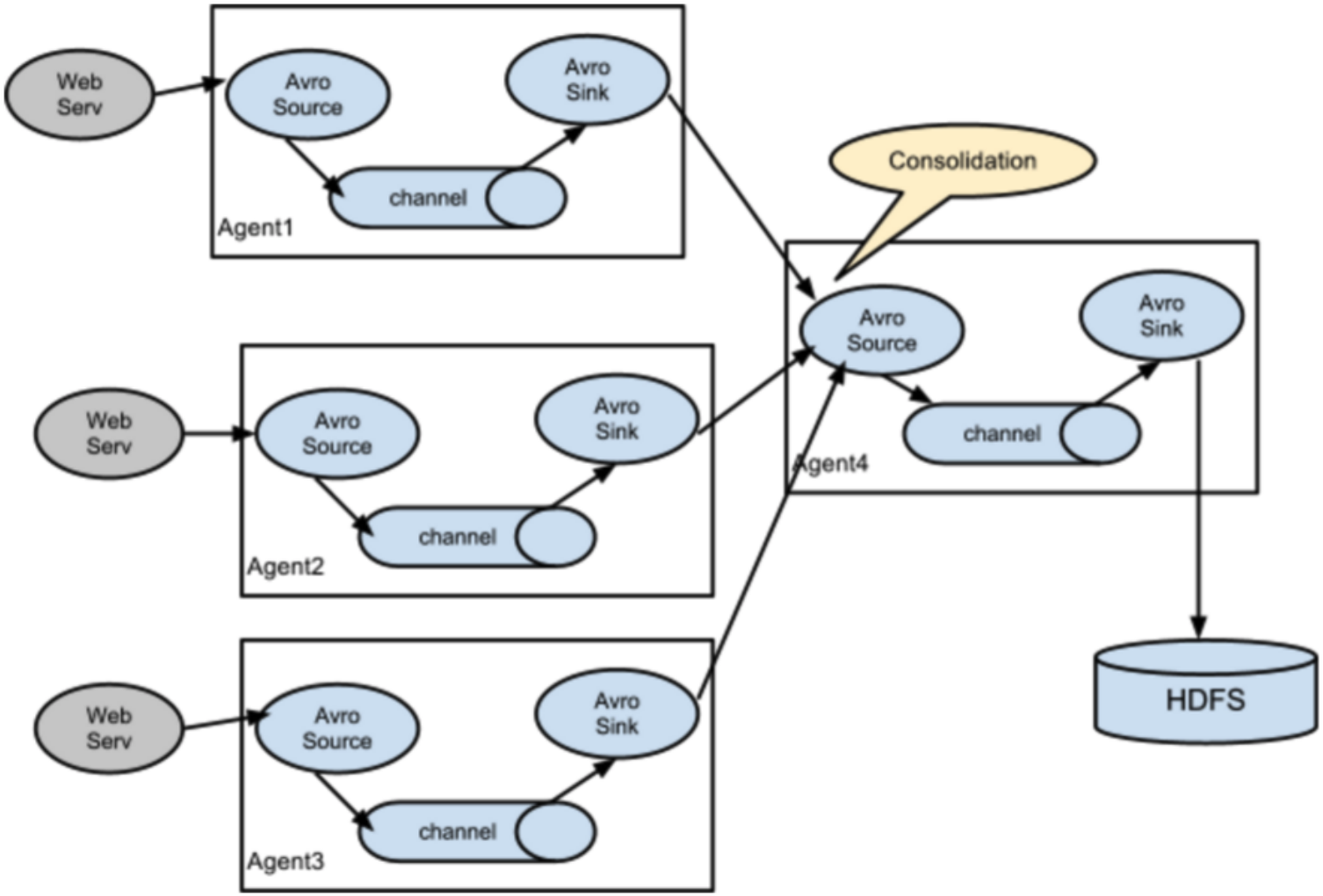
- Start Agent

```
$ flume-ng agent --conf conf --conf-file example.conf --
name a1 -Dflume.root.logger=INFO,console
```

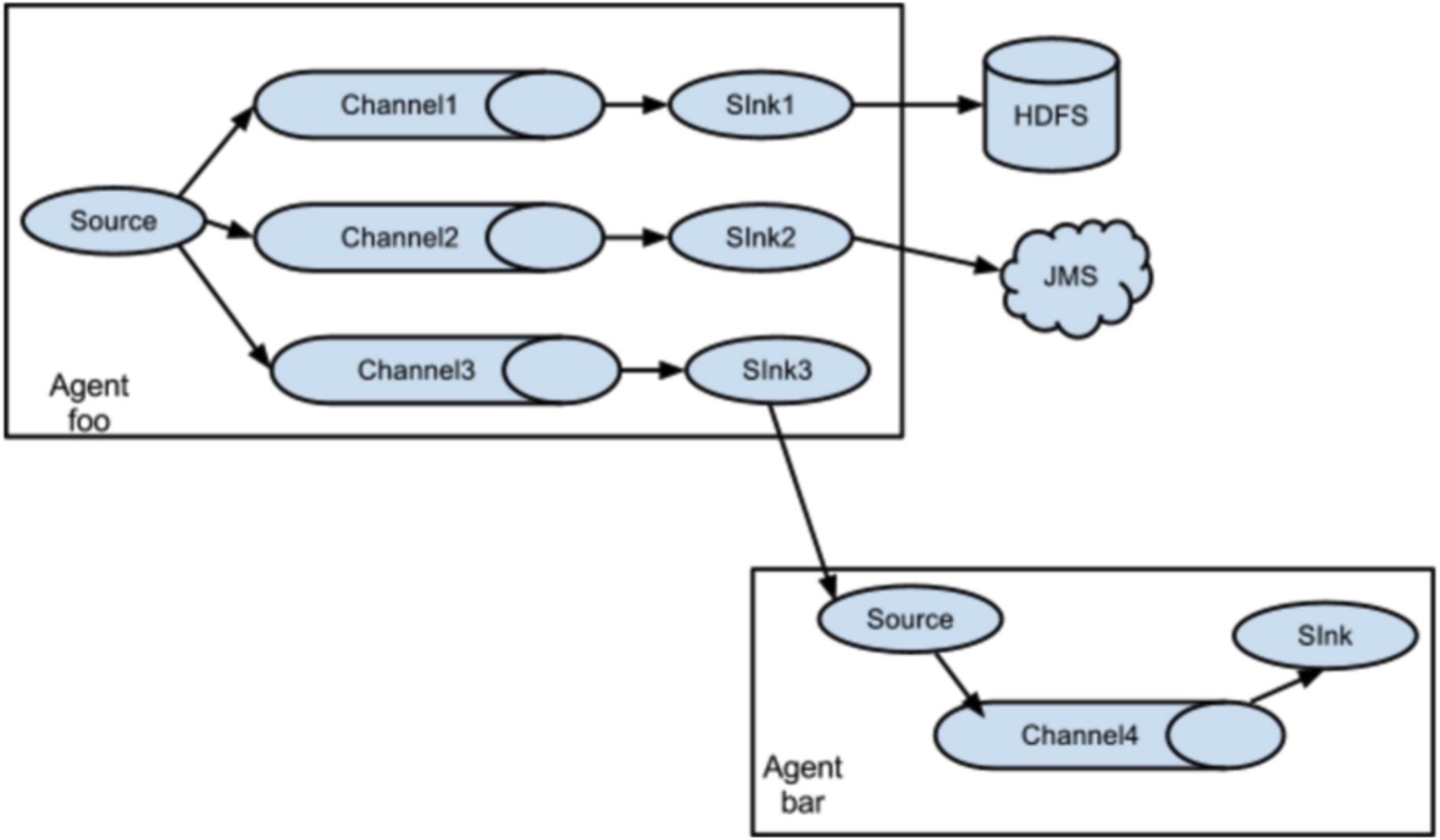
Multi-agent flow



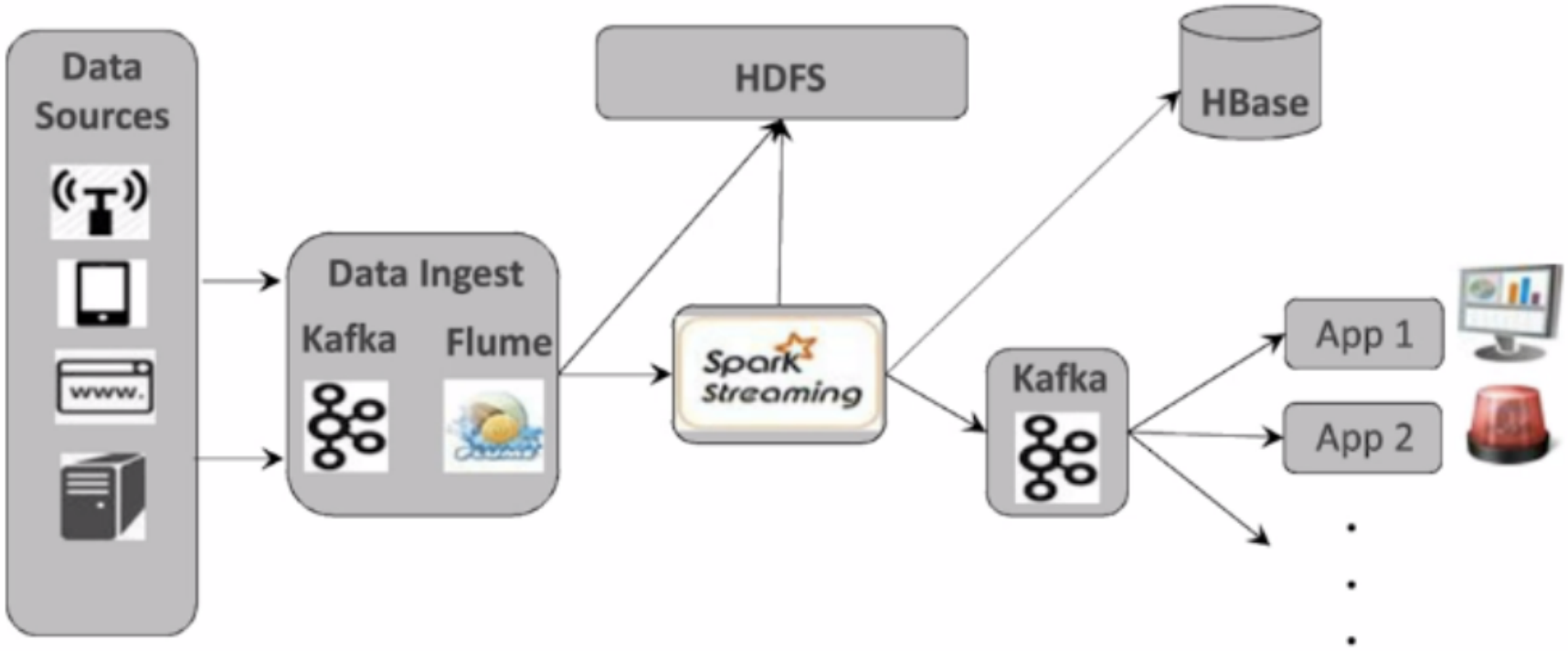
Consolidation



Multiplexing the flow



Stream Processing Architecture



Hands-On: Ingest streaming data using Flume

(LAB 1)

Preparing environment

```
# cd

# mkdir flume

# cd flume

# mkdir conf

# cd conf

# yum install nano

# 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

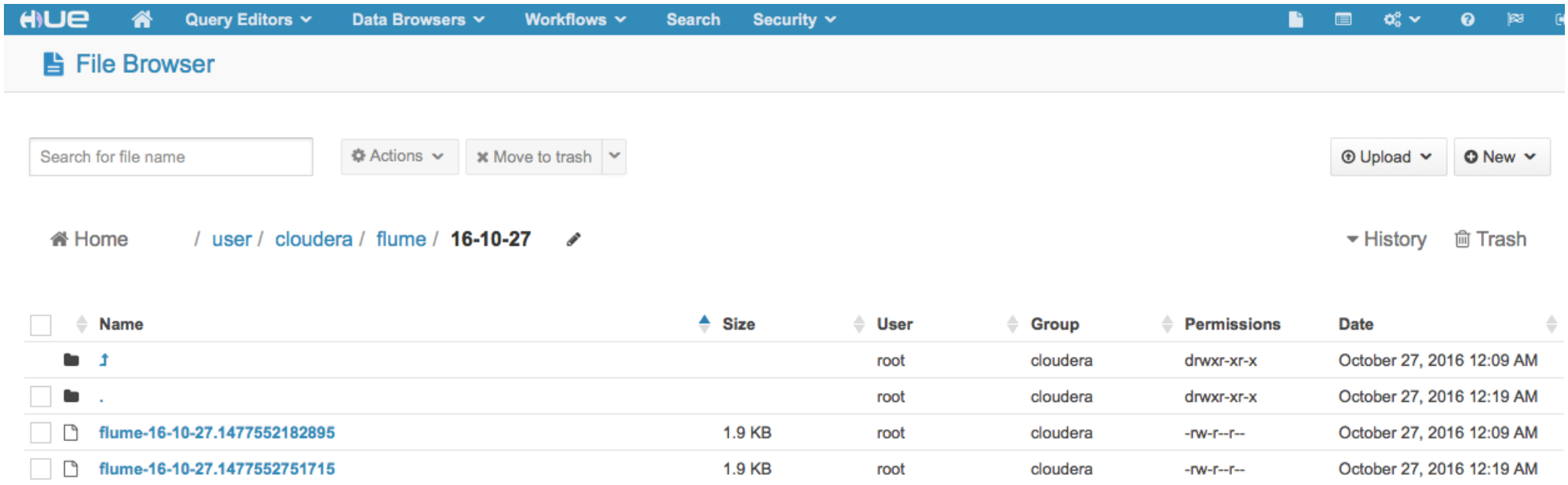
Change a permission of Hadoop directory

sudo -u hdfs hadoop fs -chmod 777 /user

flume-ng agent --name a1 --conf /root/flume/conf --conf-file /root/flume/conf/example.conf

```
16/10/27 07:19:23 INFO hdfs.BucketWriter: Closing idle bucketWriter /user/cloude
ra/flume/16-10-27/flume-16-10-27.1477552751715.tmp at 1477552763737
16/10/27 07:19:23 INFO hdfs.BucketWriter: Closing /user/cloudera/flume/16-10-27/
flume-16-10-27.1477552751715.tmp
16/10/27 07:19:23 INFO hdfs.BucketWriter: Renaming /user/cloudera/flume/16-10-27
/flume-16-10-27.1477552751715.tmp to /user/cloudera/flume/16-10-27/flume-16-10-2
7.1477552751715
16/10/27 07:19:23 INFO hdfs.HDFSEventSink: Writer callback called.
```

View a result using Hue



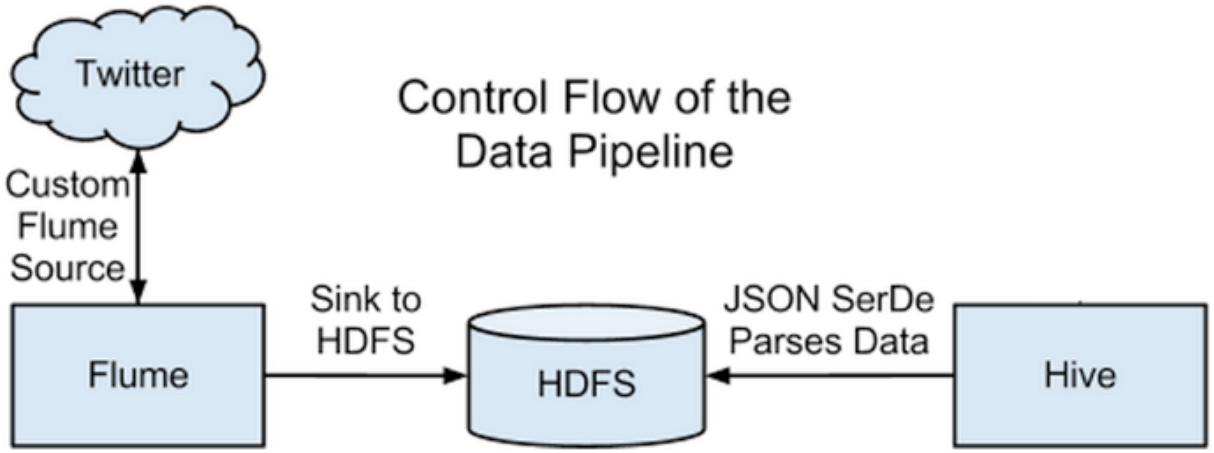
Home / user / cloudera / flume / 16-10-27 / **flume-16-10-27.1477552182895** Page 1 of 1

```
192.87.175.186 - - [01/Aug/2014:11:51:44 -0400] "GET /departments HTTP/1.1" 503 1572 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36"
195.231.2.207 - - [01/Aug/2014:11:51:45 -0400] "GET /department/fitness/products HTTP/1.1" 200 515 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:30.0) Gecko/20100101 Firefox/30.0"
65.62.183.244 - - [01/Aug/2014:11:51:46 -0400] "GET /departments HTTP/1.1" 200 756 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36"
89.92.128.155 - - [01/Aug/2014:11:51:47 -0400] "GET /departments HTTP/1.1" 200 1226 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4
) AppleWebKit/537.77.4 (KHTML, like Gecko) Version/7.0.5 Safari/537.77.4"
30.100.199.8 - - [01/Aug/2014:11:51:48 -0400] "GET /departments HTTP/1.1" 200 768 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4)
AppleWebKit/537.77.4 (KHTML, like Gecko) Version/7.0.5 Safari/537.77.4"
69.47.246.76 - - [01/Aug/2014:11:51:49 -0400] "GET /department/fitness/categories HTTP/1.1" 200 311 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:30.0) Gecko/20100101 Firefox/30.0"
71.82.19.241 - - [01/Aug/2014:11:51:50 -0400] "GET /product/291 HTTP/1.1" 200 458 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv
:30.0) Gecko/20100101 Firefox/30.0"
178.64.216.6 - - [01/Aug/2014:11:51:51 -0400] "GET /department/apparel/products HTTP/1.1" 200 741 "-" "Mozilla/5.0 (Macintosh; Intel M
ac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36"
211.60.231.72 - - [01/Aug/2014:11:51:52 -0400] "GET /product/567 HTTP/1.1" 200 2024 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4
) AppleWebKit/537.77.4 (KHTML, like Gecko) Version/7.0.5 Safari/537.77.4"
```

Hands-On: Loading Twitter Data to Hadoop HDFS

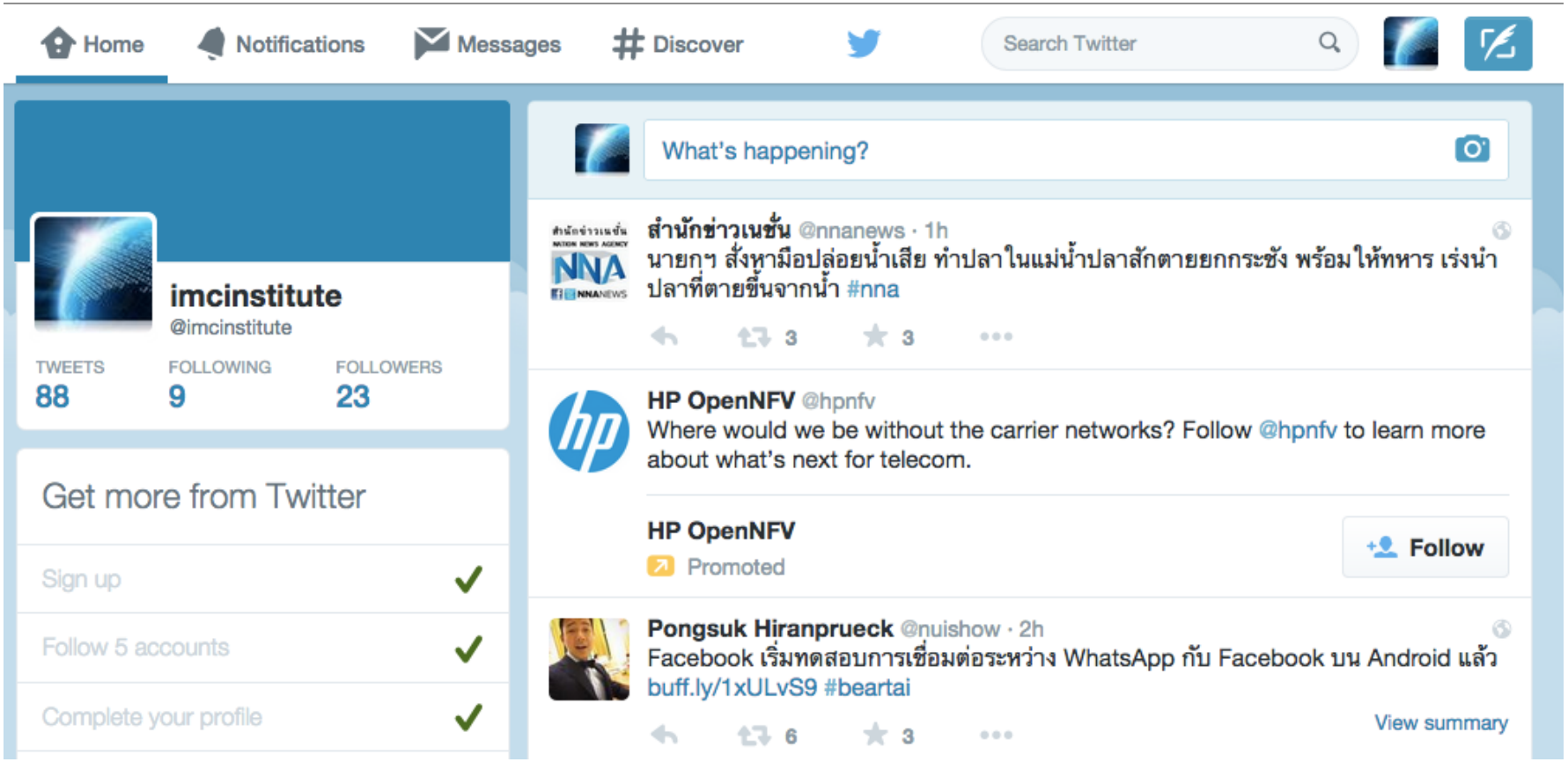
(LAB 2)

Exercise Overview



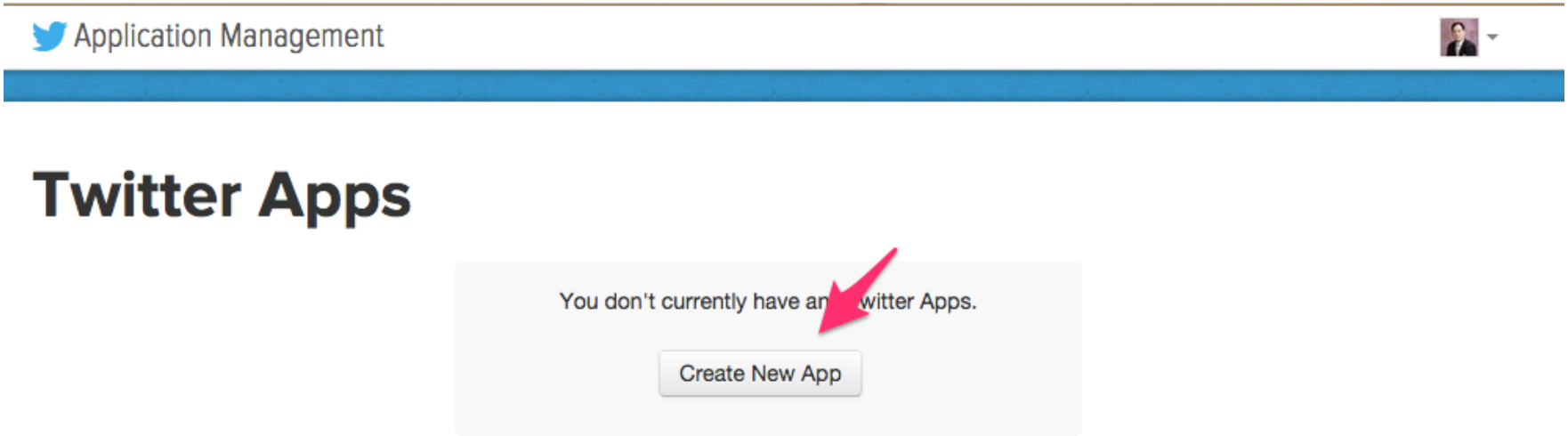
Create a new Twitter App

Login to your Twitter @ twitter.com



Create a new Twitter App (cont.)

Create a new Twitter App @ apps.twitter.com



Create a new Twitter App (cont.)

Enter all the details in the application:

Application Management

Create an application

Application Details

Name *

IMC_Institute_App

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

IMC Institute Demo App

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://www.imcinstitute.com

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the 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.)

Create a new Twitter App (cont.)

Your application will be created:

https://apps.twitter.com/app/8158163

Application Management

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

IMC_Institute_App

Test OAuth

Details | Settings | Keys and Access Tokens | Permissions

IMC Institute Demo App

http://www.imcinstitute.com

Organization

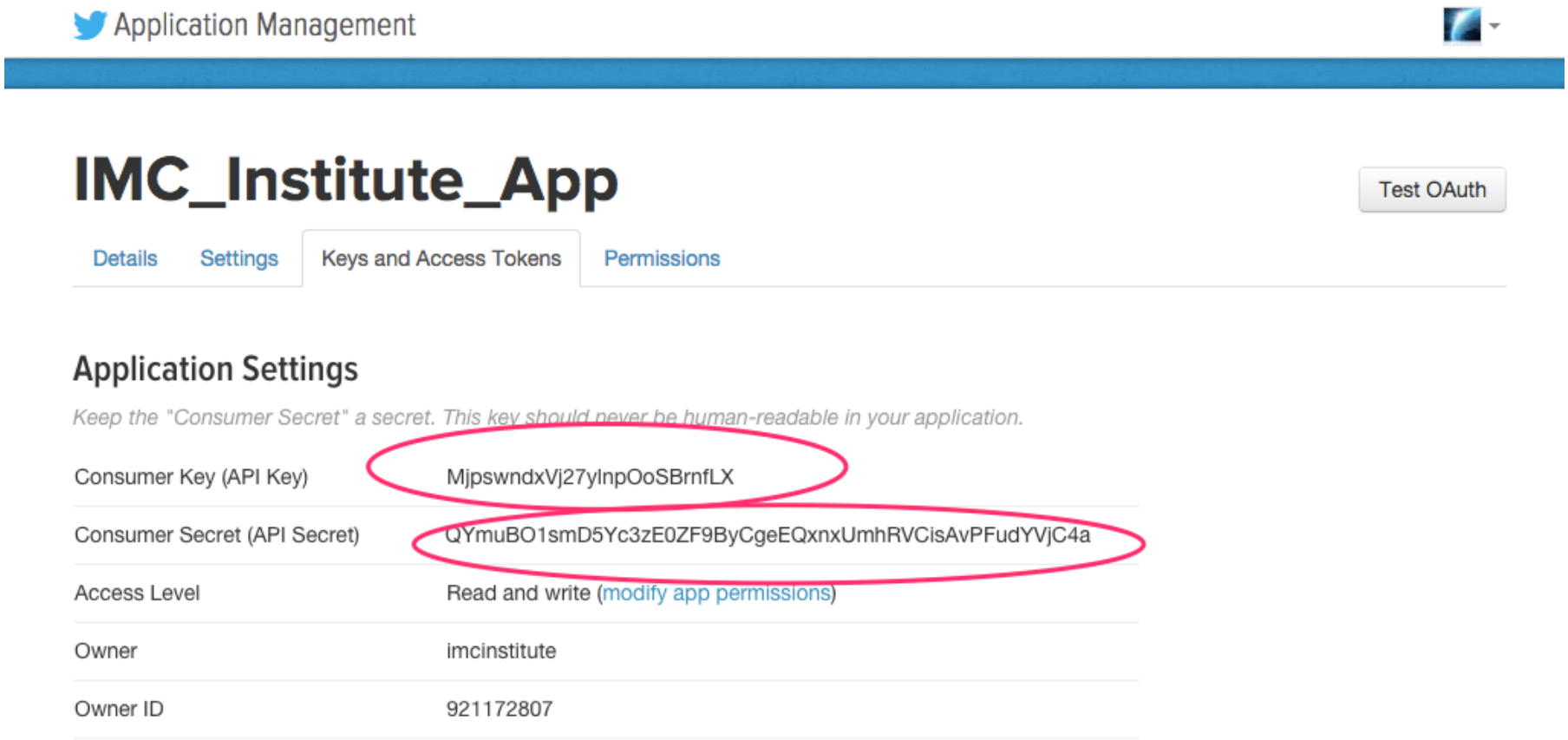
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application Settings

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	921172807-EfMXJj6as2dFECDH1vDe5goyTHcxPrF1RlJozqgx
Access Token Secret	HbpZEVip3D5j80GP21a37HxA4y10dH9BHcgEFXUNcA9xy
Access Level	Read and write
Owner	imcinstitute
Owner ID	921172807

Token Actions

Regenerate My Access Token and Token Secret | Revoke Token Access

Preparing environment

```
# cd /root/flume/conf

# rm example.conf

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

# mv example2.conf example.conf

# nano example.conf
```

Agent Configuration

```
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 =
MjpswndxVj27ylnpOoSBrnfLX
TwitterAgent.sources.Twitter.consumerSecret =
QYmuBO1smD5Yc3zE0ZF9ByCgeEQxnxUmhRVCisAvPFudYVjC4a
TwitterAgent.sources.Twitter.accessToken = 921172807-
EfMXJj6as2dFECDH1vDe5goyTHcxPrF1RIJozqgx
TwitterAgent.sources.Twitter.accessTokenSecret =
HbpZEVip3D5j80GP21a37HxA4y10dH9BHcgEFXUNcA9xy
```

Agent Configuration

```
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
```

Running Flume command

```
# flume-ng agent --name TwitterAgent --conf /root/flume/conf --conf-file
/root/flume/conf/example.conf
```

View a result using Hue