Hortonworks Data Platform

Apache Flume Component Guide

(April 20, 2017)

docs.cloudera.com

Hortonworks Data Platform: Apache Flume Component Guide

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

List of Tables

1. Introduction

Flume is a top-level project at the Apache Software Foundation. While it can function as a general-purpose event queue manager, in the context of Hadoop it is most often used as a log aggregator, collecting log data from many diverse sources and moving them to a centralized data store.

The following information is available in this chapter:

1.1. Flume Concepts

Flume Components

A Flume data flow is made up of five main components: Events, Sources, Channels, Sinks, and Agents:

Events An event is the basic unit of data that is moved using Flume. It is similar to a message in JMS and is generally small. It is made up of headers and a byte-array body.

Sources The source receives the event from some external entity and stores it in a channel. The source must understand the type of event that is sent to it: an Avro event requires an Avro source.

Channels A channel is an internal passive store with certain specific characteristics. An in-memory channel, for example, can move events very quickly, but does not provide persistence. A file-based channel provides persistence. A source stores an event in the channel where it stays until it is consumed by a sink. This temporary storage lets source and sink run asynchronously.

Sinks The sink removes the event from the channel and forwards it to either to a destination, like HDFS, or to another agent/dataflow. The sink must output an event that is appropriate to the destination.

Agents An agent is the container for a Flume data flow. It is any physical JVM running Flume. An agent must contain at least one source, channel, and sink, but the same agent can run multiple sources, sinks, and channels. A particular data flow path is set up through the configuration process.

1.2. HDP and Flume

Flume ships with many source, channel, and sink types. The following types have been thoroughly tested for use with HDP:

Sources

- Exec (basic, restart)
- Syslogtcp
- Syslogudp
- TailDir
- Kafka

Channels

- Memory
- File
- Kafka

Sinks

- HDFS: secure, nonsecure
- HBase
- Kafka
- Hive

See the Apache Flume 1.5.2 documentation for a complete list of all available Flume components.

1.3. A Simple Example

The following snippet shows some of the kinds of properties that can be set using the properties file. For more detailed information, see the Apache Flume 1.5.2 documentation.

```
agent.sources = pstream
agent.channels = memoryChannel
agent.channels.memoryChannel.type = memory

agent.sources.pstream.channels = memoryChannel
agent.sources.pstream.type = exec
agent.sources.pstream.command = tail -f /etc/passwd

agent.sinks = hdfsSinkagent.sinks.hdfsSink.type = hdfs
agent.sinks.hdfsSink.channel = memoryChannel
agent.sinks.hdfsSink.hdfs.path = hdfs://hdp/user/root/flumetest
agent.sinks.hdfsSink.hdfs.fileType = SequenceFile
agent.sinks.hdfsSink.hdfs.writeFormat = Text
```

The source here is defined as an exec source. The agent runs a given command on startup, which streams data to `stdout`, where the source gets it.

In this case, the command is a Python test script. The channel is defined as an in-memory channel and the sink is an HDFS sink.

1.4. Flume Feature Updates

Apache Flume version 1.5.2 includes cumulative 1.6 features. The table below indicates the features added to Flume 1.5.2 with each release of Hortonworks Data Platform (HDP).

Table 1.1. Apache Flume Features by HDP Version

HDP Release	Added Features	Advantages
2.5.0	Kafka Channel	Uses a single Kafka topic. Provides greater reliability and better performance.
	TailDir Source	Greater data reliability, even with rotating file names. Can restart tailing at the point where Flume stopped, while continuing data ingest.
2.4.0	Kafka Source	Reads messages from a Kafka topic. Can have multiple Kafka sources running and configure them to read a unique set of partitions for the topic.
	Kafka Sink	Publishes data to a Kafka topic. Supports pull-based processing from various Flume sources.
2.3.0	Hive Sink	Not recommended for use in production. Streams events containing delimited text or JSON data directly into a Hive table or partition. Provides a preview feature and not.

1.5. Flume 1.5.2 Documentation

See the complete Apache Flume 1.5.2 documentation for details about using Flume with Hortonworks Data Platform.

The Flume 1.5.2 documentation is also included with the Flume software. You can access the documentation in the Flume `/docs` directory.