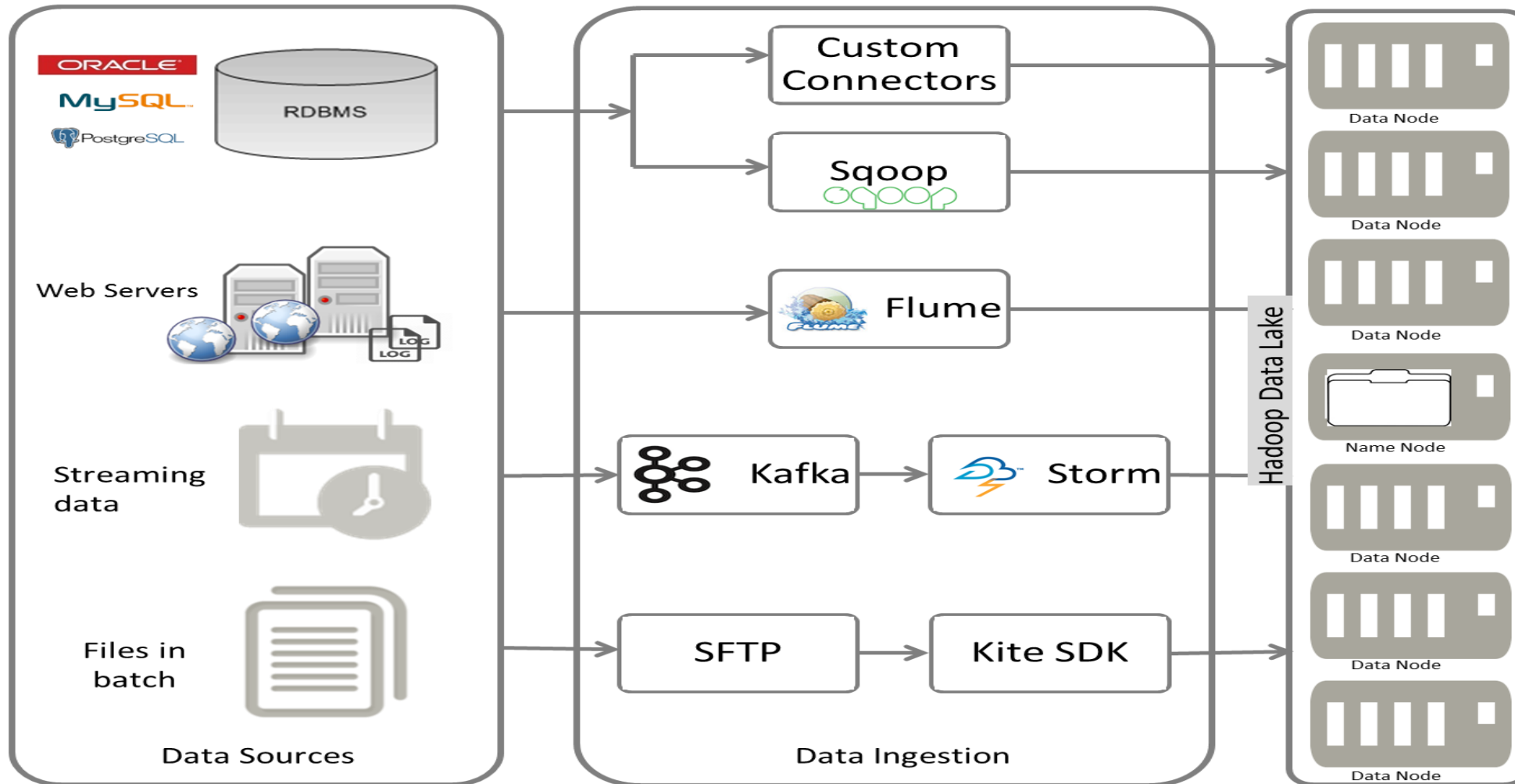


Data Ingestion



WHAT IS APACHE FLUME?

1. Apache Flume is a distributed, reliable, and available system that is used for efficiently collecting, aggregating, and moving large amounts of log data from many different sources to a centralized data store.
2. It is based on stream-oriented data flow technology which collects logs from all specified servers and loads them on central storage, such as Hadoop's Hadoop distributed file system.

HISTORY OF APACHE FLUME

Let us see a year by year evaluation of Apache Flume.

2011: Flume was first introduced in Cloudera's CDH3 distribution.

2011: In June, Cloudera moved control of the Flume project to the Apache Foundation.

2012: Refactor of Flume was done under the Star-Trek-themed tag, Flume-NG(Flume the Next Generation).

E COR OBJECTIVE OF APACHE FLUME (1/2)

Apache Flume is a Big Data tool that is designed to present a shared, secure, and highly available system to collect data from the different source systems and perform the aggregation on that and then send those data to a centralized system.

Apache Flume introduces the following summary of concepts.

Event: An event is a representation of data that is transferred by Flume from its origin to its destination.

Flow: The movement of events from the point of origin to their final destination is considered a data flow, or simply flow.

Client: It is an interface implementation that operates at the point of origin of events and delivers them to a Flume agent.

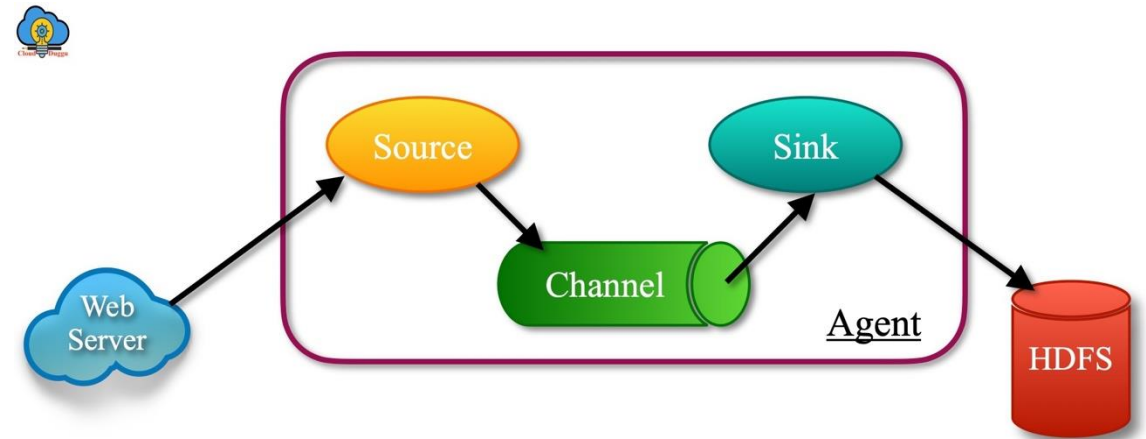
CORE OBJECTIVE OF APACHE FLUME (2/2)

Agent: It is an independent process that hosts flume components such as sources, channels, and sinks, and thus can receive, store and forward events to their next-hop destination.

Source: It is an interface implementation that can consume events delivered to it via a specific mechanism. If we take an example of Avro source that receives Avro events from the source system.

Channel: It is a transient to store the upcoming event. The events are present in the channel unless the sink consumes it.

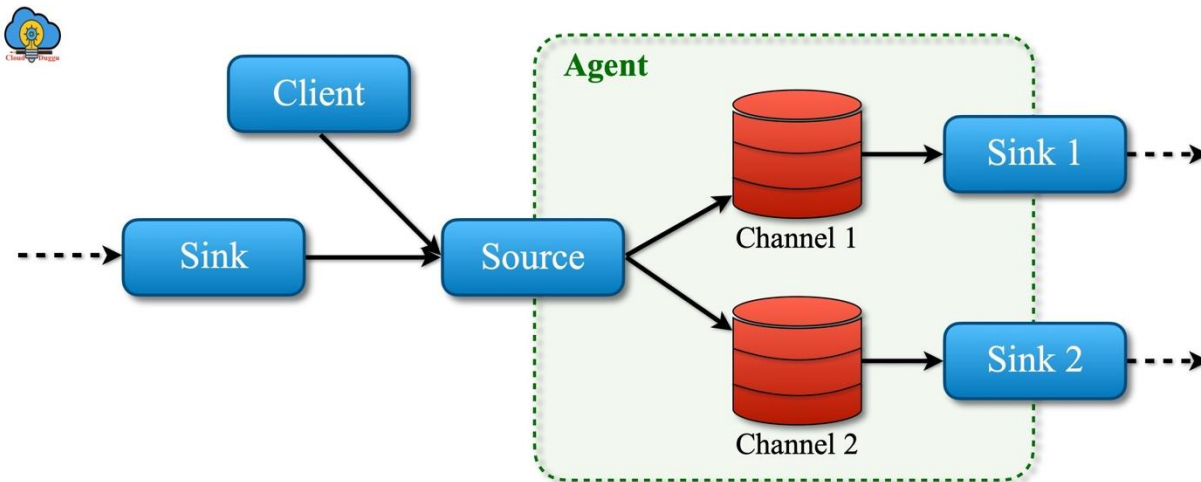
Sink: It is an interface implementation that can remove events from a channel and transmit them to the next agent in the flow, or the event's final destination.



FLOW PIPELINE OF APACHE FLUME

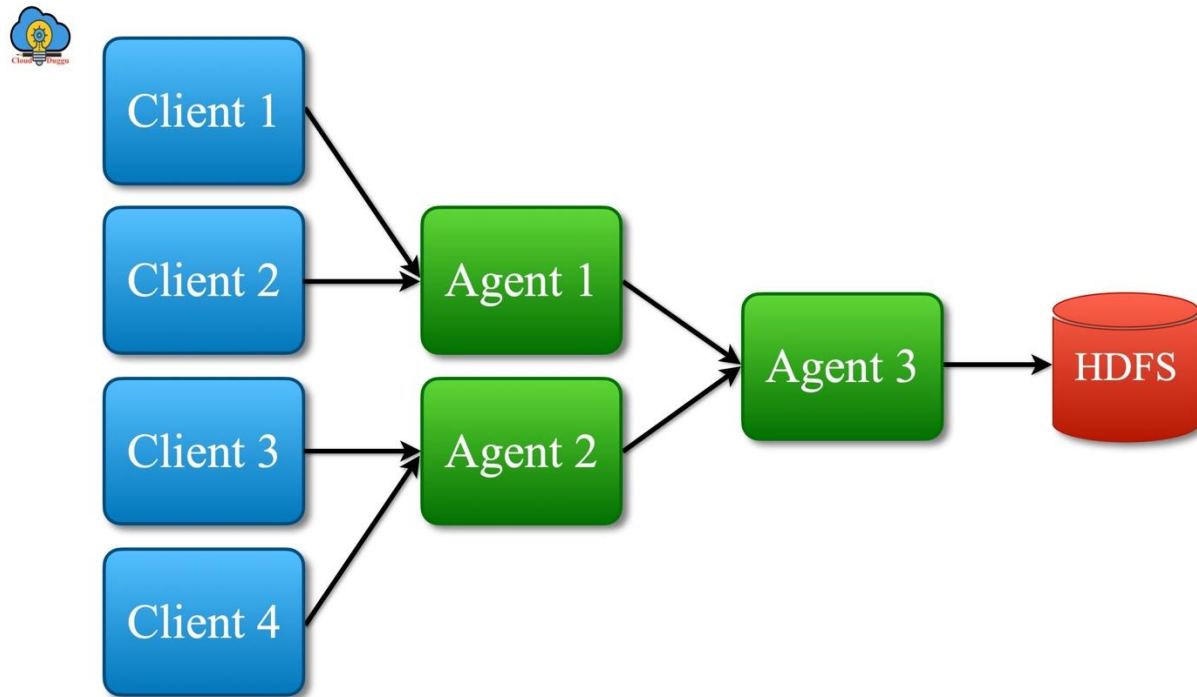
- In Apache Flume a flow pipeline starts from the client that transmits the event to its next-hop destination which is an agent. The agent will receive the event generated from the client and transfer it to more or multiple channels.
- The sinks will consume data from channels and deliver it to the next destination. In case the sink type is regular then it will send the event to another agent and in case the sink is terminal type then it will send the events to the destination.
- If there is a configuration to send an event to multiple channels then flows can fan-out to more than one destination in which the source would write the event to two channels namely Channel 1/2.
- The following figure shows, how the various components of Apache Flume interact with each other within a flow pipeline.

FLOW PIPELINE OF APACHE FLUME (1/2)



The figure shows, how the various components of Apache Flume interact with each other within a flow pipeline.

FLOW PIPELINE OF APACHE FLUME (2/2)



Equally the flows can be joined by having multiple sources operating within the same agent write to the same channel.

The following figure shows the physical layout of the converging flow.

FEATURES OF APACHE FLUME (1/2)

The following is a list of some of the important features of Apache Flume:

- Apache Flume is used to ingest data from multiple web servers in centralized storage such as Hadoop HDFS and Hbase.
- Apache Flume uses channel-based transactions to guarantee reliable message delivery.
- In case of failure, it can transmit logs without loss.
- Apache Flume provides an easy-System to add and remove agents.

FEATURES OF APACHE FLUME (2/2)

- Apache Flume's data flow is based on stream-oriented data flow in which data is transferred from source to destination through a series of nodes.
- Apache Flume allows building multi-hop fan-in and fan-out flows, contextual routing, and backup routes (fail-over) for failed hops.
- Apache Flume is also used to import a huge volume of events data such as data generated by Twitter, Facebook, Amazon, Walmart.
- Apache Flume provides high throughput with low latency.

ADVANTAGE OF APACHE FLUME (1/2)

Apache Flume provides below a list of advantages.

- Flume can be easily scaled, customized, and provides a fault-tolerant and reliable features for the multiple source system and the sinks.
- Flume can scale horizontally.
- Flume provides a stable flow of data between reading and writes operations even if the read rate exceeds the write rate.

ADVANTAGE OF APACHE FLUME (2/2)

- For each message delivery, two transactions (one sender and one receiver) are maintained.
- Flume is helpful to ingest data from a variety of sources such as network traffic, social media, email messages, log files in HDFS.
- Flume can be used to ingest data from a variety of servers in Hadoop.

DISADVANTAGE OF APACHE FLUME

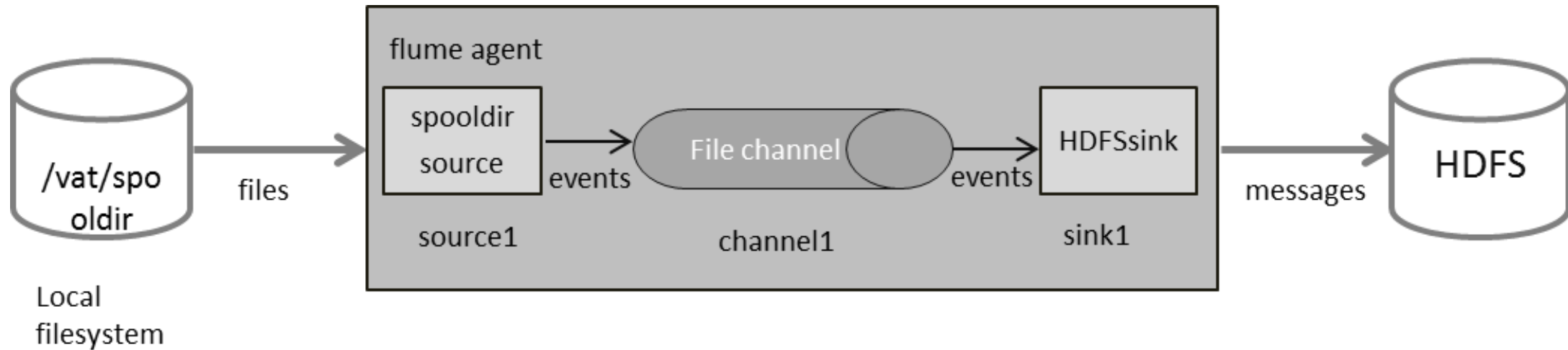
- Apache Flume architecture can become complex and difficult to manage and maintain when streaming data from multiple sources to multiple destinations.
- Flume's data streaming is not 100% real-time. Its alternatives like Kafka can be used if more real-time data streaming is needed.
- The other challenge with Apache Flume is a delicacy as it sends duplicate data as well from source to target system that is a difficult task to identify.

Flume

Flume is for high-volume ingestion into Hadoop of event-based data

e.g collect logfiles from a bank of web servers, then
move log events from those files to HDFS (clickstream)

Flume Example



Flume configuration

```
agent1.sources = source1
agent1.sinks = sink1
agent1.channels = channel1
```

```
agent1.sources.source1.channels = channel1
agent1.sinks.sink1.channel = channel1
```

```
agent1.sources.source1.type = spooldir
agent1.sources.source1.spoolDir = /var/spoolDir
```

```
agent1.sinks.sink1.type = hdfs
agent1.sinks.sink1.hdfs.path = hdfs://hostname:8020/user/flume/logs
agent1.sinks.sink1.hdfs.filetype = DataStream
```

```
agent1.channels.channel1.type = memory
agent1.channels.channel1.capacity = 10000
agent1.channels.channel1.transactionCapacity = 100
```

```
flume-ng agent -name agent1 -conf $FLUME_HOME/conf
```

| Category | Component |
|----------|--------------------|
| Source | Avro |
| | Exec |
| | HTTP |
| | JMS |
| | Netcat |
| | Sequence generator |
| | Spooling directory |
| | Syslog |
| | Thrift |
| | Twitter |
| Sink | Avro |
| | Elasticsearch |
| | File roll |
| | HBase |
| | HDFS |
| | IRC |
| | Logger |
| | Morphline (Solr) |
| | Null |
| | Thrift |
| Channel | File |
| | JDBC |
| | Memory |

Example: Using Apache Flume to Collect and Store Logs in HDFS

Step 1: Flume Configuration File (flume.conf)

This file defines the configuration for the Flume agent.

Name the components of the agent

agent1.sources = logSource

agent1.sinks = hdfsSink

agent1.channels = memoryChannel

Configure the source (e.g., monitoring a local file for new log entries)

agent1.sources.logSource.type = exec

agent1.sources.logSource.command = tail -F /var/log/system.log

agent1.sources.logSource.channels = memoryChannel

Configure the channel (memory-based)

agent1.channels.memoryChannel.type = memory

agent1.channels.memoryChannel.capacity = 10000

agent1.channels.memoryChannel.transactionCapacity = 1000

Configure the sink (writing to HDFS)

agent1.sinks.hdfsSink.type = hdfs

agent1.sinks.hdfsSink.hdfs.path =
hdfs://localhost:9000/user/flume/logs/

agent1.sinks.hdfsSink.hdfs.fileType = DataStream

agent1.sinks.hdfsSink.hdfs.writeFormat = Text

agent1.sinks.hdfsSink.hdfs.rollInterval = 60

This configuration means the HDFS sink will close the current file and open a new one every 60 seconds

agent1.sinks.hdfsSink.channel = memoryChannel

Explanation of the Configuration:

1.Source: Uses the exec type to tail a log file (/var/log/system.log) in real-time.

2.Channel: A memory channel temporarily buffers events between the source and sink.

3.Sink: Sends the ingested data to an HDFS directory (/user/flume/logs/).

Example: Using Apache Flume to Collect and Store Logs in HDFS

- **Step 2: Start the Flume Agent**

- Run the Flume agent using the command:

- **`flume-ng agent --conf ./conf --conf-file flume.conf --name agent1 -Dflume.root.logger=INFO,console`**

- Replace `./conf` with the path to your Flume configuration directory.

- The agent will now begin ingesting log data into HDFS.

- **Step 3: Verify Data in HDFS**

- Once the agent is running, you can verify that the data is being written to HDFS by listing the target directory:

- **`hdfs dfs -ls /user/flume/logs/`**

- You should see files containing the log entries being ingested.

- This simple setup demonstrates how Apache Flume can efficiently collect, aggregate, and move large amounts of log data into a distributed storage system like HDFS.