Collect Web Server Logs with Flume

Exercise Path: ~/workspace/flume

Local Data Path: ~/materials/data/weblogs/

**In this exercise, you will configure Flume to ingest web log data from a local directory to HDFS.**

Apache web server logs are generally stored in files on the local machines running the server. In this exercise, you will simulate an Apache server by placing provided web log files into a local spool directory, and then use Flume to collect the data.

Both the local and HDFS directories must exist before using the spooling directory source.

**Creating an HDFS Directory for Flume Ingested Data**

1. Create a directory in HDFS called lanier/weblogs to hold the data files Flume ingests:

$ hdfs dfs -mkdir -p lanier/weblogs

**Creating a Local Directory for Web Server Log Output**

1. Create the spool directory into which our web log simulator will store data files for Flume to ingest. On the local filesystem create

flume/weblogs\_spooldir:

$ sudo mkdir -p flume/weblogs\_spooldir

1. Give all users the permissions to write to the flume/weblogs\_spooldir directory:

$ sudo chmod a+w -R flume

**Configuring Flume**

In workspace/flume under fixme edit the provided Flume starter configuration file with the characteristics listed below. If you need help getting started, you may refer to configuration file in hints, or view the solution in the solutions directory.

Have students copy the contents of the “solutions” folder to the “studs” version of weblogs\_spooldir. (See step:5 for ultimate solution)

* The source is a spooling directory source that pulls from

flume/weblogs\_spooldir

* The sink is an HDFS sink that:
  + Writes files to the lanier/weblogs directory
  + Disables time-based file rolling by setting the hdfs.rollInterval property to 0
  + Disables event-based file rolling by setting the hdfs.rollCount property to 0
  + Sets the hdfs.rollSize property as 524288 to enable size-based file rolling at 512Html\_site
  + Writes raw text files (instead of SequenceFile format) by setting hdfs.fileType to DataStream
* The channel is a Memory Channel that:
  + Can store 10,000 events using the capacity property
  + Has a transaction capacity of 1,000 events using the transactionCapacity property

**Flume Performance Note**

Flume’s performance on a VM will vary. Sometimes, Flume will exhaust the

memory capacity of its channel and give the following error:

Space for commit to queue couldn't be acquired. Sinks are likely not

keeping up with sources, or the buffer size is too tight.

If you get this error, you will need to increase the capacity and

transactionCapacity properties to a higher value. Then, you will need to

restart the agent, clean up any files in HDFS, and repeat the Flume

operation.

**Running the Agent**

After you have created the configuration file, start the agent and copy the files to the spooling directory.

1. Change directories:

$ cd materials/data/exercises/flume

1. Start Flume by using the configuration you just made. Replace solution with fixme to run your own configuration:

$ flume-ng agent --conf /etc/flume-ng/conf

--conf-file *solution*/spooldir.conf

--name agent1 -Dflume.root.logger=INFO,console

1. Wait a few moments and the Flume agent to start up. You should see a message like:

Component type: SOURCE, name: webserver-log- source started

**Simulating Apache Web Server Output**

1. Open a separate terminal window, and change to the exercise directory. Run the script to place the web log files in the /flume/weblogs\_spooldir directory:

$ cd materials/data/exercises/flume

$ ./copy-move-weblogs.sh /flume/weblogs\_spooldir

This script will create a temporary copy of the web log files and move them to

the spooldir directory.

1. Return to the terminal that is running the Flume agent and watch the logging output. The output will give information about the files Flume is putting into HDFS.
2. Once the Flume agent has finished, enter CTRL+C to terminate the process.

NOTE: That the files that were imported are tagged with a Unix timestamp corresponding to the time the file was imported, such as

FlumeData.1427214989392.

**END**