cloudera®

Ask Bigger Questions

CAP - Developing with Spark and Hadoop:

Homework Assignment Guide for Students

Homework: Collect Web Server Logs with Flume

Files and Data Used in this Homework

Exercise directory: `$DEV1/exercises/flume`

Data (local): `$DEV1DATA/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.

Create an HDFS directory for Flume to save ingested data

1. Create a directory in HDFS called `/loudacre/weblogs` to hold the data files Flume ingests, e.g.

```
$ hdfs dfs -mkdir /loudacre/weblogs
```

Create a local directory for web server log output

2. 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
```

3. Give all users the permissions to write to the `/flume/weblogs_spooldir` directory:

```
$ sudo chmod a+w -R /flume
```

Configure Flume

In `$DEV1/exercises/flume` create a Flume 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 (professor VM only).

- The source is a spooling directory source that pulls from `/flume/weblogs_spooldir`
- The sink is an HDFS sink that:
  - Writes files to the `/loudacre/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 512KB
  - 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 10,000 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.

Run the Agent

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

4. Change directories to the `$DEV1/exercises/flume` directory.

5. Start the Flume agent using the configuration you just made. For example, in the professor VM, if you wish to use the solution configuration, enter this command (replace `solution` with `stubs` 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
```

6. Wait a few moments for the Flume agent to start up. You will see a message like: `Component type: SOURCE, name: webserver-log-source started`

Simulate Apache web server output

7. 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 $DEV1/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.

8. 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.

9. Once the Flume agent has finished, enter `CTRL+C` to terminate the process.

10. Using the hdfs command line or Hue File Browser, list the files in HDFS that were added by the Flume agent.

    Note that the files that were imported are tagged with a Unix timestamp corresponding to the time the file was imported, e.g. `FlumeData.1427214989392`

Optional Homework

Loudacre has a data source that comes in via the network. You want to see some examples since the format is undocumented.

Flume has a `netcat` source, which allows Flume to listen on a port and process the contents received. The address to bind to is specified by the `bind` property, and the port to bind to is specified by the `port` property.

Flume also has a `logger` sink, which logs any incoming data at the `INFO` level in Log4J. This is helpful for debugging and testing Flume configuration. This sink does not require any configuration properties.

Create a Flume configuration file with the following characteristics:

- Uses the `netcat` source to capture data from host `localhost` on port `12345`.
- Uses the `logger` sink
- Uses the `memory` channel

Start the Flume agent using the new configuration file.

In another terminal window, start the test data feed script using the command:

```
$ $DEV1/exercises/flume/script/rng_feed.py
```

Observe the log messages in the Flume agent's terminal and stop the agent after you have seen a few example records.

This is the end of the Homework