Session 4: Lab Documents for Flume Code Run on AWS EMR

Step 1: Open the terminal and change to root user using the following command:

```
sudo -i
```

Configure the Flume agent to consolidate the logs from the spoolDir source to an Avro sink. Create spool.conf to configure the agent using the following command:

```
vi spool.conf
```

Press i and insert the following:

```
#list sources, sinks and channels in the agent
spoolagent.sources = spoolsrc
spoolagent.channels = sachnl
spoolagent.sinks = avrosnk

# define the flow
spoolagent.sources.spoolsrc.channels = sachnl
spoolagent.sinks.avrosnk.channel = sachnl

# source type properties
spoolagent.sources.spoolsrc.type = spooldir
spoolagent.sources.spoolsrc.spoolDir = /tmp/UpGradLogs

# sink type and properties
spoolagent.sinks.avrosnk.type = avro
spoolagent.sinks.avrosnk.bind=localhost
spoolagent.sinks.avrosnk.hostname = localhost
spoolagent.sinks.avrosnk.port = 12345
spoolagent.sinks.avrosnk.batch-size = 100
spoolagent.sinks.avrosnk.rollCount = 0
spoolagent.sinks.avrosnk.rollInterval = 0
spoolagent.sinks.avrosnk.rollSize = 100000

# channel type and properties
```

```
spoolagent.channels.sachnl.type = file
```

Press escape and type :**wq!** to save and exit.

Configure the Flume agent to consolidate the logs from the Avro source to an HDFS sink. Create hdfssink.conf to configure the Flume agent using the following command:

```
vi hdfssink.conf
```

Press i and insert the following:

```
#list sources, sinks and channels in the agent
hdfsagent.sources = avrosrc
hdfsagent.sinks = hdfssnk
hdfsagent.channels = hachnl

# define the flow
hdfsagent.sources.avrosrc.channels = hachnl
hdfsagent.sinks.hdfssnk.channel = hachnl

# source type properties
hdfsagent.sources.avrosrc.type = avro
hdfsagent.sources.avrosrc.bind = localhost
hdfsagent.sources.avrosrc.hostname = localhost
hdfsagent.sources.avrosrc.port = 12345
hdfsagent.sources.avrosrc.batch-size = 100
hdfsagent.sources.avrosrc.rollCount = 0
hdfsagent.sources.avrosrc.rollInterval = 0
hdfsagent.sources.avrosrc.rollSize = 100000

# sink type and properties
hdfsagent.sinks.hdfssnk.type = hdfs
hdfsagent.sinks.hdfssnk.hdfs.path = /test/flume/data

# channel type and properties
hdfsagent.channels.hachnl.type = memory
hdfsagent.channels.hachnl.capacity = 100000
hdfsagent.channels.hachnl.transactionCapacity = 100000
```

Press escape and type :**wq!** to save and exit.

Step 2: On the terminal with the root user, remove the 'UpGradLogs' and 'test' folders if they already exist using the following command:

```
rm -rf /tmp/UpGradLogs
hadoop fs -rm -r /test
```

UpGradLogs is a folder inside /tmp. This will be generated when you run the LogGenerator.jar, and it will contain the logs. To avoid any conflict ahead, you need to remove the folder. Similarly, test will be a folder inside HDFS where the ingested logs will be dumped; so, you need to remove it too in order to avoid any conflict ahead.

The final logs will reside at **/test/flume/data**. Let's check whether any data is currently in this directory using the following command:

```
hadoop fs -ls /test/flume/data
```

You will see that the folder does not exist.

Step 3: On the terminal with the root user, download and run the JAR file to generate logs at /tmp/UpgradLogs using the following commands:

```
wget -P /root/LogGeneratorProcess
https://s3.amazonaws.com/upgradflumedata/LogGenerator.jar
```

```
chmod 755 /root/LogGeneratorProcess/LogGenerator.jar
java -jar /root/LogGeneratorProcess/LogGenerator.jar
```

Step 4: Open a new terminal, change to root user (**sudo -i**), and run these commands to see the list of log files generated after the JAR has completed running:

```
ls /tmp/UpGradLogs
```

So, you have the logs available at this location ready to be consumed.

Step 5: Open a new terminal, change to root user (**sudo -i**), and run the log collection Flume agent using the following command:

```
flume-ng agent -n spoolagent -c conf -f spool.conf
-Dflume.root.logger=INFO,console
```

This Flume agent will consolidate the logs from the spoolDir source to an Avro sink.

Step 6: Open a new terminal, change to root user (**sudo -i**), and run the HDFS Flume agent using the following command:

```
flume-ng agent -n hdfsagent -c conf -f hdfssink.conf
-Dflume.root.logger=INFO,console
```

This Flume agent will now consolidate the logs from the Avro source to an HDFS sink.

Step 7: Open a new terminal, change to root user (**sudo -i**), and check the Hadoop file directory using the following command:

```
hadoop fs -ls /test/flume/data
```

So now, you have your data at the **/test/flume/data** folder. You can see the content of any of the files using **cat**.

```
hadoop fs -cat /test/flume/data/*
```

You can see how the logs residing at the /tmp/UpGradLogs folder are ingested inside HDFS using the two Flume agents. Moreover, the logs files at /tmp/UpGradLogs that are successfully ingested are given a '.completed' suffix by Flume, which helps the Flume agent identify which files not to ingest again.

Note: In production systems, such logs are generated continuously, and Flume agents continuously keep on ingesting the data from the production systems to the Hadoop clusters. For example, you can modify the LogGenerator.jar to continuously generate logs at the /tmp/UpGradLogs folder, and the same Flume agent that you configured will serve the purpose of continuously ingesting the files from the /tmp/UpGradLogs folder to HDFS.

Step 8: Press **Ctrl-C** to exit the processes on each terminal.