Hadoop and Spark Developer - CCA 175

Problem Scenario 7 [FLUME]

CCA 175 Hadoop and Spark Developer Exam Preparation - Problem Scenario 7

PLEASE READ THE INTRODUCTION TO THIS SERIES. CLICK ON HOME LINK AND READ THE INTRO BEFORE ATTEMPTING TO SOLVE THE PROBLEMS

Video walkthrough of this problem is available at [*CLICK HERE*]

Click here for the video version of this series. This takes you to the youtube playlist of videos.

This question focusses on validating your ***flume*** skills. You can either learn flume by following the video accompanied with this post or learn flume elsewhere and then solve this problem while using the video as a reference. This video serves both as tutorial and walkthrough of how to leverage flume for data ingestion.

Note: While this post only provides specifics related to solving the problem, the video provides an introduction, explanation and more importantly application of flume knowledge.

Problem 7:

1. This step comprises of three substeps. Please perform tasks under each subset completely
   - using sqoop pull data from MYSQL **orders** table into /**user/cloudera/problem7/prework** as **AVRO** data file using only one mapper
   - Pull the file from **user\cloudera\problem7\prework** into a local folder named **flume-avro**
   - create a flume agent configuration such that it has an avro source at localhost and port number 11112, a jdbc channel and an hdfs file sink at **/user/cloudera/problem7/sink**
   - Use the following command to run an avro client **flume-ng avro-client -H localhost -p 11112 -F <<Provide your avro file path here>>**
2. The CDH comes prepackaged with a log generating job. **start_logs, stop_logs** and **tail_logs**. Using these as an aid and provide a solution to below problem. The generated logs can be found at path **/opt/gen_logs/logs/access.log**
   - run start_logs
   - write a flume configuration such that the logs generated by start_logs are dumped into HDFS at location **/user/cloudera/problem7/step2.** The channel should be non-durable and hence fastest in nature. The channel should be able to hold a maximum of **1000** messages and should commit after every **200** messages.
   - Run the agent.
   - confirm if logs are getting dumped to hdfs.
   - run stop_logs.

Solution:

Step 1:

Pull orders data from order sqoop table to **\user\cloudera\problem7\prework**

```
sqoop import --table orders --connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" --username retail_dba --password cloudera -m 1 --target-dir /user/cloudera/problem7/prework --as-avrodatafile
```

Get the file from HDFS to local

```
mkdir flume-avro;
cd flume-avro;
hadoop fs -get /user/cloudera/problem7/prework/* .
gedit f.config
```

Create a flume-config file in problem7 folder named **f.config**

```
#Agent Name = step1

# Name the source, channel and sink
step1.sources = avro-source
step1.channels = jdbc-channel
step1.sinks = file-sink

# Source configuration
step1.sources.avro-source.type = avro
step1.sources.avro-source.port = 11112
step1.sources.avro-source.bind = localhost


# Describe the sink
step1.sinks.file-sink.type = hdfs
step1.sinks.file-sink.hdfs.path = /user/cloudera/problem7/sink
step1.sinks.file-sink.hdfs.fileType = DataStream
step1.sinks.file-sink.hdfs.fileSuffix = .avro
step1.sinks.file-sink.serializer = avro_event
step1.sinks.file-sink.serializer.compressionCodec=snappy

# Describe the type of channel -- Use memory channel if jdbc channel does not work
step1.channels.jdbc-channel.type = jdbc

# Bind the source and sink to the channel
step1.sources.avro-source.channels = jdbc-channel
step1.sinks.file-sink.channel = jdbc-channel
```

Run the flume agent

```
flume-ng agent --name step1 --conf . --conf-file f.config
```

Run the flume Avro client

Search This Blog

Search

CCA 175 Hadoop and Spark Developer Preparation

- Home
- CCA 175 Prep Plan
- Problem Scenario 1
- Problem Scenario 2
- Problem Scenario 3
- Problem Scenario 4
- Problem Scenario 5 [SQOOP]
- Problem Scenario 6 [Data Analysis]
- **Problem Scenario 7 [FLUME]**
- File Formats
- Youtube Playlist

A leader with a unique blend of deep technology expertise and strong management skil

Arun Kumar Pasuparthi

Follow 212

View my complete profile

Report Abuse

Blog Archive

April 2017 (1)

```
flume-ng avro-client -H localhost -p 11112 -F <<Provide your avro file path here>>
```

Step 2:

```
mkdir flume-logs
cd flume-logs
```

create flume configuration file

```
# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F /opt/gen_logs/logs/access.log


# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.hdfs.path = /user/cloudera/problem7/step2
a1.sinks.k1.hdfs.fileSuffix = .log
a1.sinks.k1.hdfs.writeFormat = Text
a1.sinks.k1.hdfs.fileType = DataStream

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 200

# Bind the source and sink to the channel
a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1
```

create hdfs sink directory

```
hadoop fs -mkdir /user/cloudera/problem7/sink
```

Run the flume-agent

```
flume-ng agent --name a1 --conf . --conf-file f.config
```

PLEASE SEE VIDEO FOR A COMPLETE WALKTHROUGH OF THIS SOLUTION

14 comments:

Deven May 24, 2017 at 12:49 PM

Nicely presented flume scenario. Please keep publishing the content on CCA175 certification. Thanks a ton!

Reply

Deven May 24, 2017 at 1:56 PM

After creating the Avro file in hdfs /user/cloudera/problem7/sink, When I tried reading the Avro file in spark i get msg saying "java.io.IOException: Not an Avro data file".
I checked the flume-ng process I see msg "17/05/24 13:41:01 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false ".

I have these parms set.
step1.sinks.file-sink.hdfs.fileType = DataStream
step1.sinks.file-sink.hdfs.fileSuffix = .avro
step1.sinks.file-sink.hdfs.serializer = avro_event

Reply

Replies

Arun Kumar Pasuparthi May 25, 2017 at 9:06 AM

Thank you Deven for posting your concern. You actually opened a can of worms here with flume 1.6. Let me explain you how to negotiate through your problem here.

The configuration you used and I showed in the video had minor error. I corrected it in the blog and will correct in in the video too.

Below is the correct configuration. notice there is no word hdfs in the key of the property.

step1.sinks.file-sink.serializer = avro_event

If you get any class cast exception using this configuration then switch to memory channel. Some people reported that jdbc channel and avro_event are not doing well with each other.

Note that above produces an avro file\files in hdfs whose schema is different than source. if you want to have the same schema as source then you will have to use a customer serializer.

Reply

Deven May 25, 2017 at 10:55 AM

Arun thanks for looking into the issue, much appreciated. Will try the options.

Reply

Sayali Mahajan May 27, 2017 at 1:56 PM

In CCA175 is there any probability that they will as Kafka and spark streaming as per new syllabus?
Any idea Arun Sir?

Reply

Replies

Arun Kumar Pasuparthi May 28, 2017 at 4:46 PM

I dont believe there would be a question on Kafka. There may be one on Spark streaming but no one has reported it so far so not sure. Please watch my video on certification preparation strategy to understand exam objective to technology mapping.

Reply

Bala Hassan June 22, 2017 at 11:35 PM

Hi Arun,

I am getting the following error for step 1, when I run an avro client:

17/06/22 23:30:19 WARN api.NettyAvroRpcClient: Using default maxIOWorkers
17/06/22 23:30:24 ERROR avro.AvroCLIClient: Unable to deliver events to Flume. Exception follows.
org.apache.flume.EventDeliveryException: NettyAvroRpcClient { host: quickstart.cloudera, port: 11112 }: Failed to send batch
at org.apache.flume.api.NettyAvroRpcClient.appendBatch(NettyAvroRpcClient.java:315)
at org.apache.flume.client.avro.AvroCLIClient.run(AvroCLIClient.java:229)
at org.apache.flume.client.avro.AvroCLIClient.main(AvroCLIClient.java:72)
Caused by: org.apache.flume.EventDeliveryException: NettyAvroRpcClient { host: quickstart.cloudera, port: 11112 }: Avro RPC call returned Status: FAILED

Could you please advise the reason for the same? I am seeing some data being sent to sink directory though.

Thanks,
Bala.

Reply

Replies

Arun Kumar Pasuparthi June 23, 2017 at 3:45 AM

try switching to a memory channel. i could not solve this problem using jdbc channel, seems like an inherent bug in the spark version we use. It is beyond my knowledge to solve this. If you notice, i also mentioned the same in the problem solution (i.e to switch to memory channel)

Bala Hassan June 23, 2017 at 7:13 AM

Hi Arun, I tried with memory channel as well, but still the same issue. Anyways, I will give it a try again and will update you in case I am able to fix it.

Thanks,
Bala.

Reply

Kanan August 1, 2017 at 10:35 AM

This comment has been removed by the author.

Reply

Kanan August 1, 2017 at 10:58 AM

This comment has been removed by the author.

Reply

Unknown October 4, 2017 at 10:21 PM

Hi Arun

Thanks for this wonderful blog and youtube session for Flume. Because of this, I could understand the concept behind usage of Flume.

Reply

GodfreyDeK November 21, 2017 at 7:06 AM

Greetings Arun,

Thank you for your work here, it's been a real blessing to us all. I've looked at the 'new syllabus' and it states skills required to ingest real-time and near-real-time data -which after reading the documentation on spark streaming, seems like a tool to use for the job. However, as you have not covered it yet in your blog series, could you be so kind as to suggesting a way (or resource) for relevant practicing of spark streaming?

Seven blessings to you.

Regards

Reply

Replies

Arun Kumar Pasuparthi November 21, 2017 at 10:21 AM

Hello Godfrey,

I am not sure if Cloudera is going to ask anything on Spark Streaming even though it is part of Syllabus. The reason i say this, setting up a scenario in such a way that a test taker can just write the spark streaming code is a very difficult job. Most importantly, even if cloudera figures out a way to setup that scenario, i am not sure if there is an automated way in which they can validate the code you wrote. For example, for all of the spark or hive or sqoop or flume configuration you write, the result is static after your finish the code and cloudera can automate the scanning of the results by running some queries. However, doing this can be extremely challenging when you have to compare streaming result. The world 'streaming' means the result keeps changing based on the data in the available stream. So i think, you can safely go into the exam and still clear it even though you dont have any working knowledge on the spark streaming. However, i would still encourage you to equip yourself in spark, flink, storm and other streaming libraries to excel in day to day nuances of being a data engineer.

Reply

Add comment

Enter your comment...

Comment as: Select profile...

Publish Preview

Home

Subscribe to: Posts (Atom)

If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

(no title)
If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

Simple theme. Powered by Blogger.