Flume - Introduction

Flume - Introduction

Flume is a simple, robust and extensible tool for data ingestion from various data sources into Hadoop. It is used for collecting, aggregating and transporting a large amount of streaming data such as events and logs from various sources to a centralized data store such as HDFS.

[Flume - Use Case]

Let's assume an e-commerce company wants to analyze customer behavior. To do so, they will need to move customer logs and events data from various sources such as web servers and databases to HDFS or HBase to perform the analysis. Here Flume can be useful to move the customer's data to HDFS.

Flume - Agents

Flume agents are independent daemon processes. Flume agent consists of three parts: Source, Channel and Sink.

- A source receives data from the data generators

Examples - Tweets, Web server logs, Click event data in any application, etc

- A channel is a transient store which receives the events from the source and buffers them till they are consumed by the sinks. It acts as a bridge between the sources and the sinks.

Examples - File System channel, memory channel, etc

- A sink consumes data from the channels and delivers it to the destination.

Examples - HDFS, HBase etc

When the rate of incoming data from the source exceeds the rate at which it can be written to the destination, flume channel acts as a mediator between the source and sink by buffering the data.

[Flume - Use Case - Agents]

Flume agents run on every machine where we want to collect the data.

As displayed in the image, in our previous use case, flume agents will be installed on every web server where data is being produced. A data collector collects data from agents and pushes it to a centralized data store.

[Slide Flume - Multiple Agents]

Flume agents can be arranged in arbitrary topologies. As shown in the image, the source is consuming data from the sink and the same sink data is getting consumed by multiple sources.

Flume - Sources & Delivery Reliability

Flume supports a wide variety of sources like tail, syslog and log4j. Tail is a Unix command which outputs the last few lines of the file. syslog is a standard for logging messages in Unix. Every application in Unix sends logs to syslog. log4j is the logging framework for Java applications.

Flume provides two types of delivery reliability - Best-effort and end-to-end. Best-effort delivery does not tolerate any node failures while end-to-end guarantees delivery even in the event of node failures.

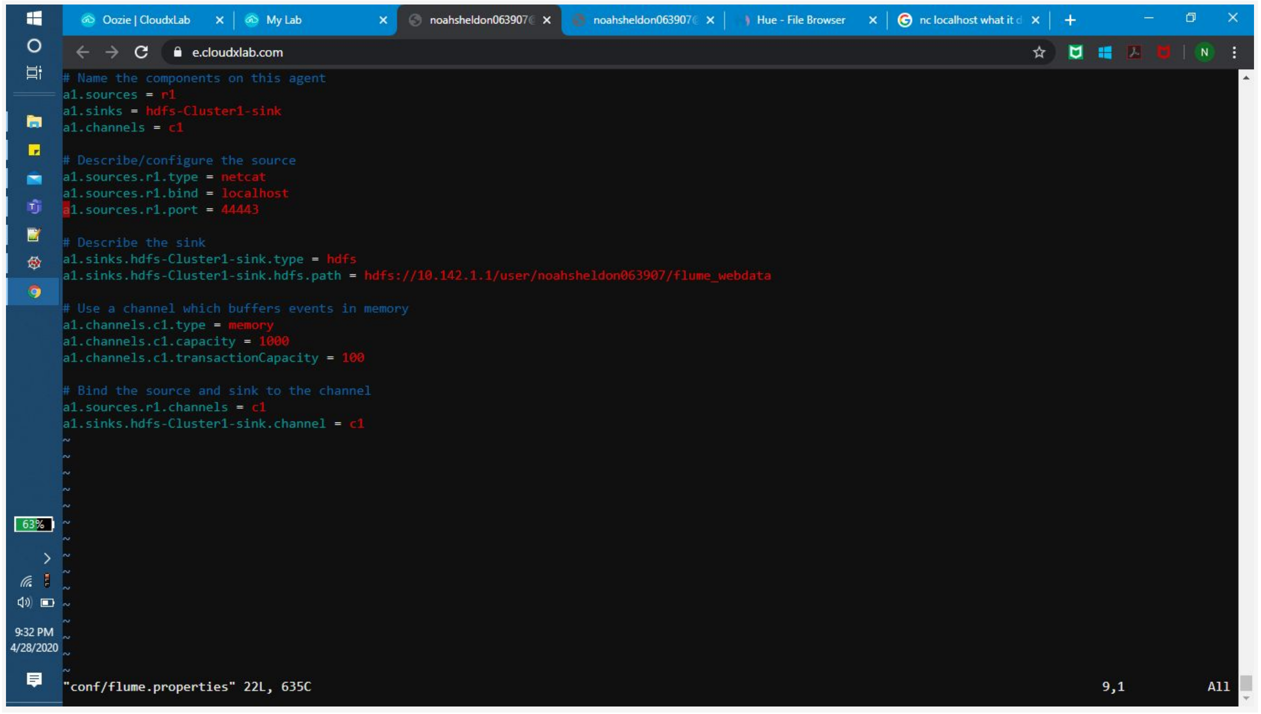
Hands-on Demo on CloudxLab

Flume - Hands-On Steps

```
#Get a copy of sample flume conf from common data
hadoop fs -copyToLocal /data/flume/conf

# Change the port if needed and location in HDFS
nano conf/flume.properties

Change the port number and userid path
```



```
# Name the components on this agent
a1.sources = r1
a1.sinks = hdfs-Cluster1-sink
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 44443

# Describe the sink
a1.sinks.hdfs-Cluster1-sink.type = hdfs
a1.sinks.hdfs-Cluster1-sink.hdfs.path = hdfs://10.142.1.1/user/noahsheldon063907/flume_webdata

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.hdfs-Cluster1-sink.channel = c1
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"conf/flume.properties" 22L, 635C                                          9,1           All
```

```
#Launch the flume agent
flume-ng agent --conf conf --conf-file conf/flume.properties --name a1
Dflume.root.logger=INFO,console


# Open a new console and Connect to the same port that you defined in config
nc localhost 44443

# Generate some data
Type something in the console

#Open a new console and Check in hdfs using
hadoop fs -ls flume_webdata
hadoop fs -cat 'flume_webdata/FlumeData*'
```

Flume - Hands-on

Let's do a hands-on exercise in Flume. We will read the data from the port and push it to HDFS. Login to the CloudxLab Linux console on two different terminals. On the first terminal, we will run flume-agent and on the second terminal, we will run a server from which we will read the data.

Copy flume configuration from HDFS to the Linux console. It is located at /data/flume/conf on HDFS. Open flume.properties. We've defined configurations for agent a1 in this file. We can define configuration for multiple agents a1, a2, a3 in the same file. While running flume, we can specify the name of the agent which we want to run on that machine.

We have specified source type as netcat. netcat is a good way to quickly create a server which listens on a specified port. Let's change the port number to 44444. While running flume-agent if port 44444 is used by any other user, it will throw up an "Address already in use" error. In that case please change the port to some other number like 44445 or 44446 in flume configuration file.

Sink type is HDFS. Change HDFS sink path to your home directory in HDFS.

Also please note that we are specifying the channel type as memory which will buffer events in memory. Bind the source and sink to the channel.

Let's run the flume-agent on the first terminal. Please note that we are specifying the agent name as a1. Port 44444 is used by another process. Let's change the port to 44445 and run the flume agent again. Port 44445 is also used by another process. Change the port to 44443. Run the agent again and this time it is started successfully

Now let’s produce some data. Go to the second terminal and type nc localhost 44443. Type in some data and see if it gets pushed to HDFS in the sink path.

Login to Hue and check the path in the File browser. Here we can see the data in binary format.

In this video, we have covered Flume introduction and its use case. we have also discussed flume agents and demonstrated steps on how to use Flume on CloudxLab.

Hope you enjoyed the video. Happy learning!

Summary

In previous videos on Flume, we have covered Flume introduction and its use case. we have also discussed flume agents and demonstrated steps on how to use Flume on CloudxLab. Hope you enjoyed the videos. Happy learning!