Experiment 11 - Streaming Data Analysis using Flume

| Roll No. |  |
| --- | --- |
| Name |  |
| Class | D15A |
| Subject | DS using Python Lab |
| LO Mapped | LO5: Design and Build an application that performs exploratory data analysis using Apache Spark |
|  |  |

**Aim**:

To perform streaming data analysis using Flume.

**Introduction**:

Big data streaming is a process in which big data is quickly processed in order to extract real-time insights from it. The data on which processing is done is the data in motion. Big data streaming is ideally a speed-focused approach wherein a continuous stream of data is processed

This data needs to be processed sequentially and incrementally on a record-by-record basis or over sliding time windows, and used for a wide variety of analytics including correlations, aggregations, filtering, and sampling. Information derived from such analysis gives companies visibility into many aspects of their business and customer activity such as –service usage (for metering/billing), server activity, website clicks, and geo-location of devices, people, and physical goods –and enables them to respond promptly to emerging situations. For example, businesses can track changes in public sentiment on their brands and products by continuously analyzing social media streams, and respond in a timely fashion as the necessity arises.

**Benefits of streaming data**

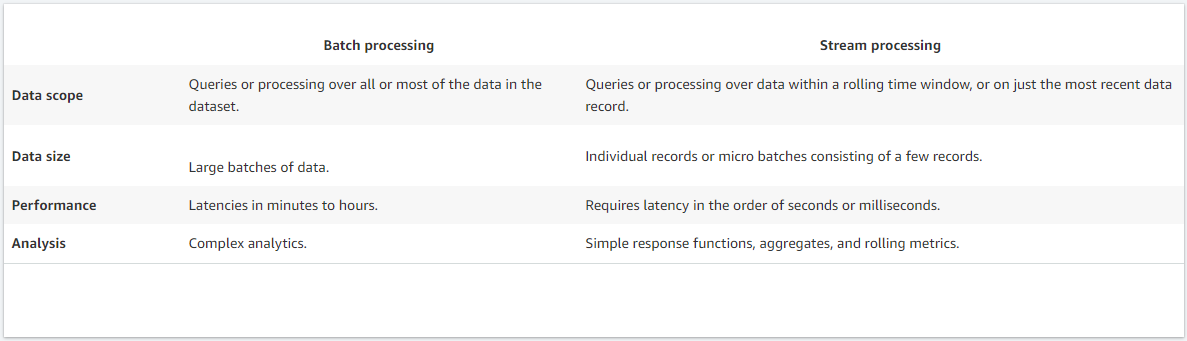
* Streaming data processing is beneficial in most scenarios where new, dynamic data is generated on a continual basis.
* It applies to most of the industry segments and big data use cases. Companies generally begin with simple applications such as collecting system logs and rudimentary processing like rolling min-max computations. Then, these applications evolve to more sophisticated near-real-time processing.
* Initially, applications may process data streams to produce simple reports, and perform simple actions in response, such as emitting alarms when key measures exceed certain thresholds.
* Eventually, those applications perform more sophisticated forms of data analysis, like applying machine learning algorithms, and extract deeper insights from the data.
* Over time, complex, stream and event processing algorithms, like decaying time windows to find the most recent popular movies, are applied, further enriching the insights.

**Disadvantages of Data Streams**

* Lack of security of data in the cloud
* Hold cloud donor subordination
* Off-premises warehouse of details introduces the probable for disconnection

**Examples of streaming data**

* A financial institution tracks changes in the stock market in real time, computes value-at-risk, and automatically rebalances portfolios based on stock price movements.
* A real-estate website tracks a subset of data from consumers’ mobile devices and makes real-time property recommendations of properties to visit based on their geo-location.
* A solar power company has to maintain power throughput for its customers, or pay penalties. It implemented a streaming data application that monitors all of panels in the field, and schedules service in real time, thereby minimizing the periods of low throughput from each panel and the associated penalty payouts.
* A media publisher streams billions of clickstream records from its online properties, aggregates and enriches the data with demographic information about users, and optimizes content placement on its site, delivering relevancy and better experience to its audience.
* An online gaming company collects streaming data about player-game interactions, and feeds the data into its gaming platform. It then analyzes the data in real-time, offers incentives and dynamic experiences to engage its players.

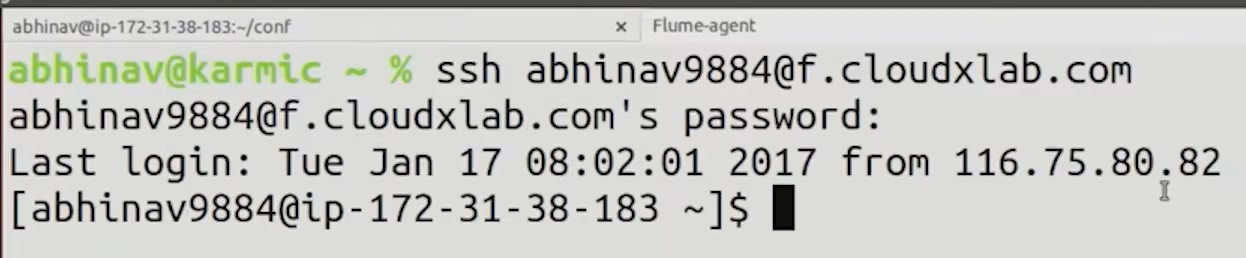


**Data Capture using Flume**

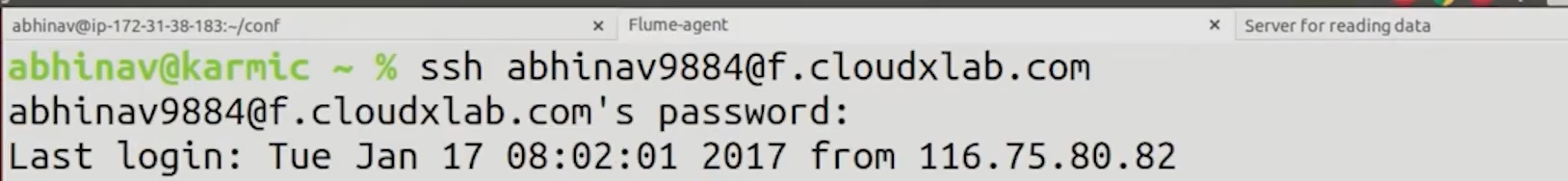
1. To read the data from the port and push it to HDFS. Login to the CloudxLab Linux console on two different terminals.
2. On the first terminal, run **flume-agent** and on the second terminal, run a server from which to **read the data**.
3. Copy flume configuration from HDFS to the Linux console.
   1. It is located at */data/flume/conf* on HDFS.
4. Open **flume.properties**. configurations for agent a1 is defined in this file. To define configuration for multiple agents a1, a2, a3 in the same file. While running flume, specify the name of the agent which you want to run on that machine.
5. Specify source type as netcat. netcat is a good way to quickly create a server which listens on a specified port.
6. Change the port number to **44443**. 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 change the port to some other number like 44445 or 44446 in the flume configuration file.
7. Sink type is HDFS. Change HDFS sink path to your home directory in HDFS.
8. Specifying the channel type as memory will buffer events in memory. Bind the source and sink to the channel.
9. To 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.
10. Now open a new console and find the list of files created by flume in hdfs using the command
    1. *hadoop fs -ls flume\_webdata*
11. To view the data in the first file, run the command
    1. *hadoop fs -cat* followed by the
    2. *flume\_webdata/the filename*
12. The filename will start with FlumeData followed by a dot and then a number. This number is the timestamp.
13. Here we can see the data which we entered earlier. If you wish to view the rest of the files, you can use -cat command with the other files in similar fashion. Please note that if the data is coming too frequently, please view the last few lines with -tail instead of viewing the entire file with -cat.

**Results**:

Run flume agent in first terminal

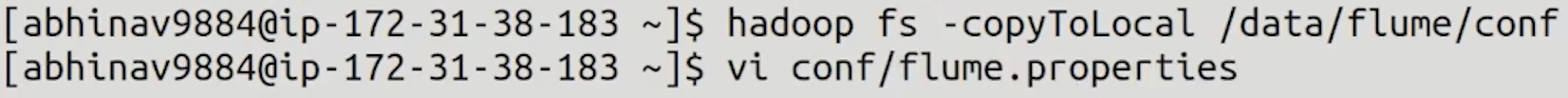
**

Run a server on second terminal, through which we will read the data



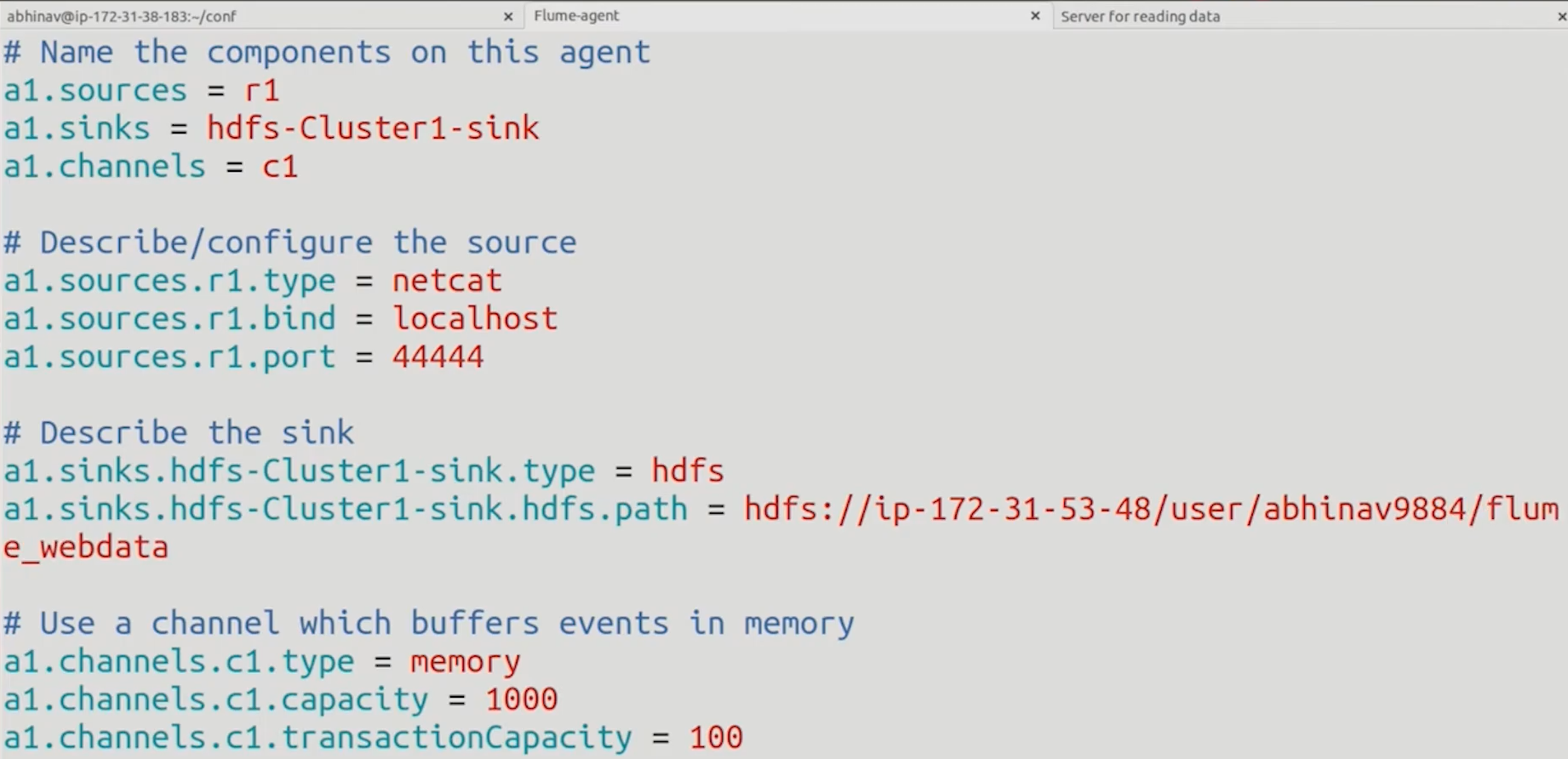
Copy flume config from HDFS to linux console using the following command

Then open flume.Properties using

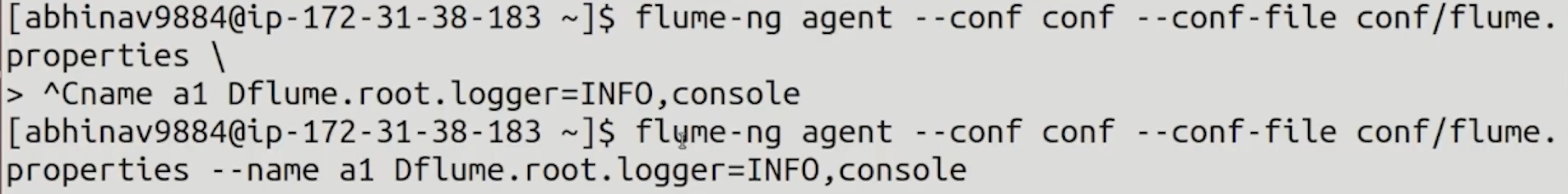


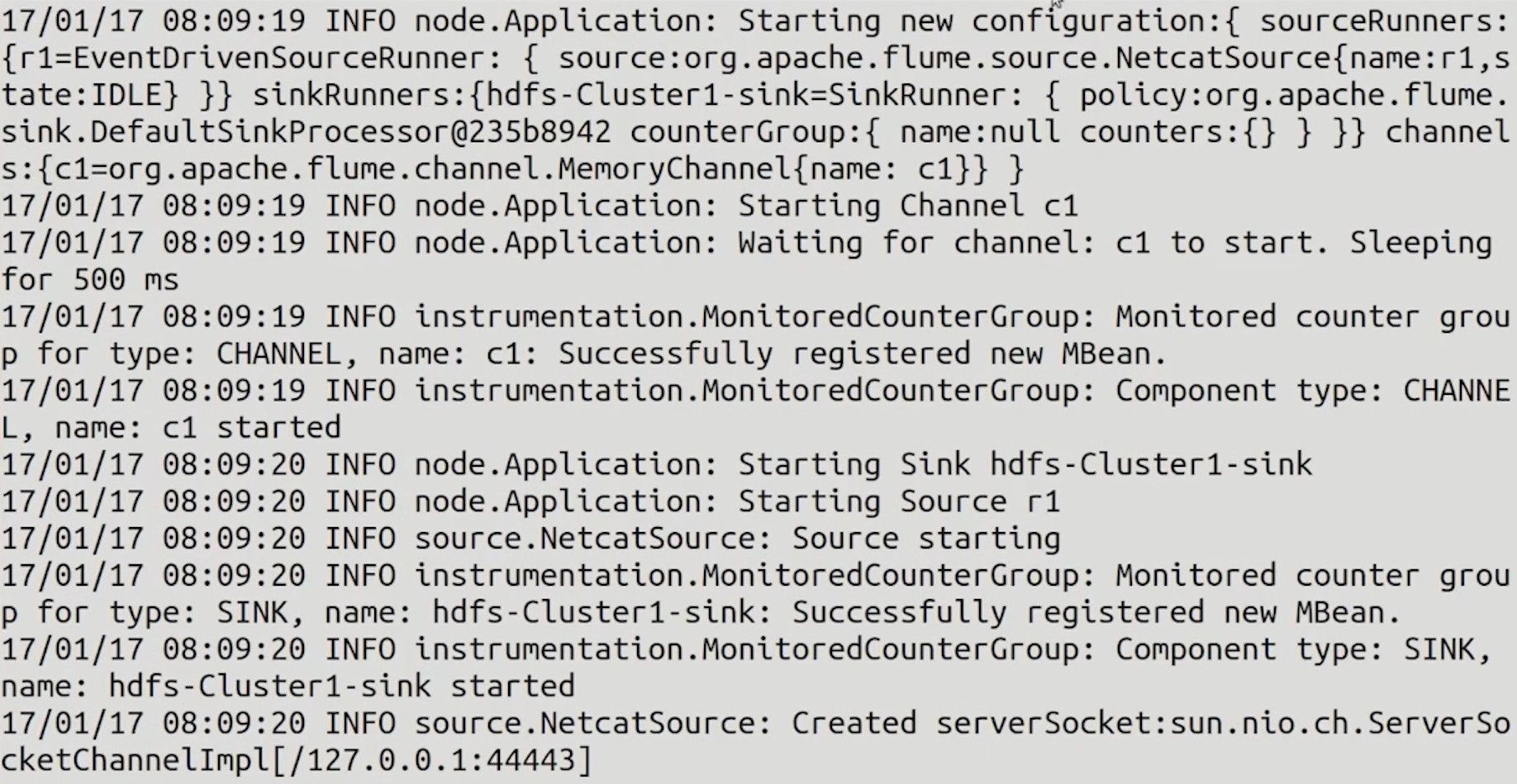
This shows configuration of agents, here we have defined agent a1

Change the port to 4443 and set a1.sinks.hdfs-Cluster1-sing.hdfs.path to the path from you home directory



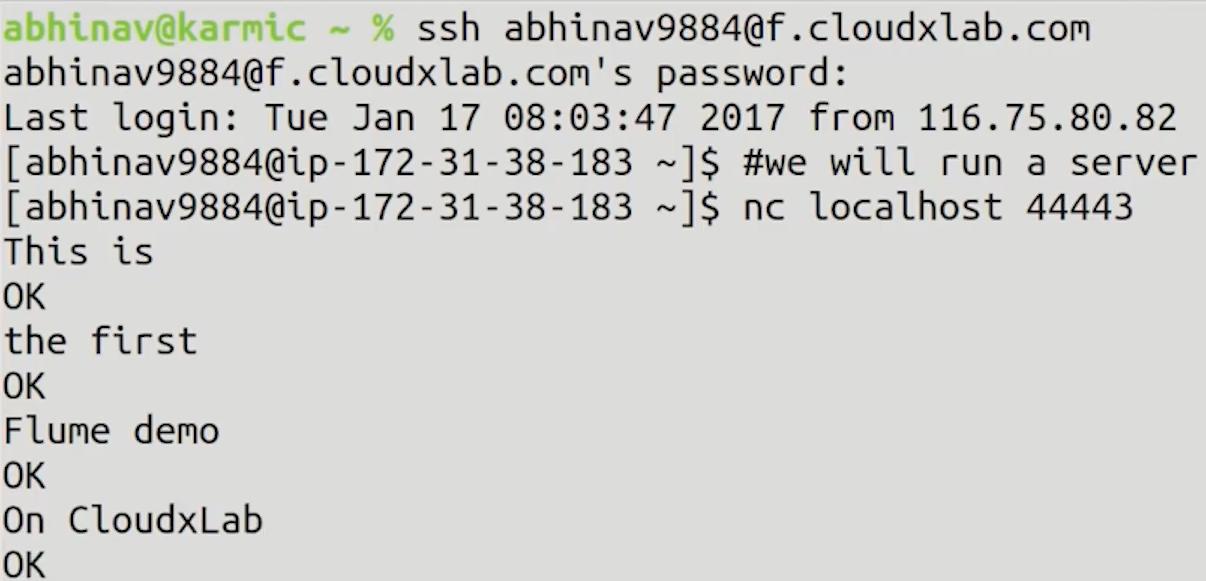
On the first terminal, run the flume agent using the following command





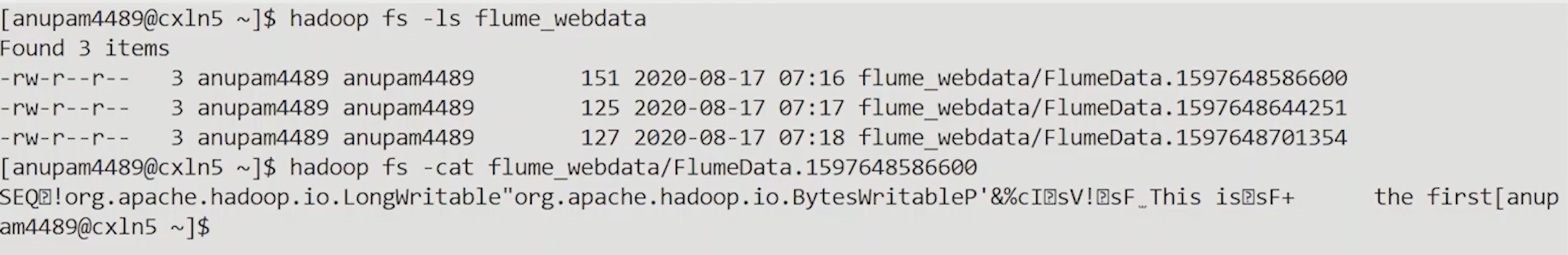
To produce some data, go to second terminal and write the following command

Then enter the data you want



We can find the list of files created by flume in HDFS using the following command

To view the data use the next command

****

**Conclusion**:

Hence we have successfully studied and understood the concept of streaming data using flume