Vishwakarma University, Pune
University Grants Commission (UGC) Approved State Private University

Assignment 3	
Name - Jyot Buch	Roll No. - 09
Date of Submission - 01/05/2023	Class - A (2023 Batch)

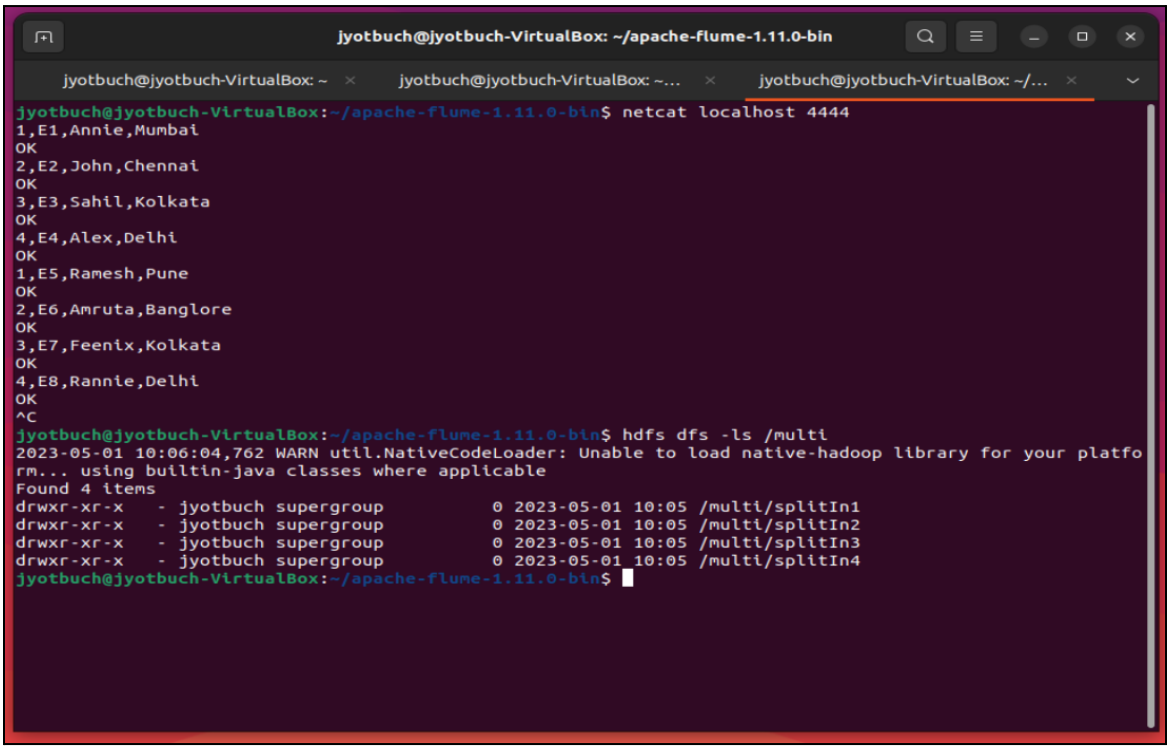
1. In the Big Data world, one of the main jobs is to collect, aggregate, and move data from a single source or many sources to a centralized data store or multiple destinations. Based on this, you are given one scenario where you are supposed to multiplex data from the terminal with two channels and two sinks. Support your answer with a screenshot of CLI fetching data from HDFS out of four separate directories. The input for your problem is given below:

```
Emp_Dep  Emp_Dep     Name       City
1,       E1,         Annie,     Mumbai
2,       E2,         John,      Chennai
3,       E3,         Sahil,     Kolkata
4,       E4,         Alex,      Delhi
1,       E5,         Ramesh,    Pune
2,       E6,         Amruta,    Banglore
3,       E7,         Feenix,    Kolkata
4,       E8,         Rannie,    Delhi
```

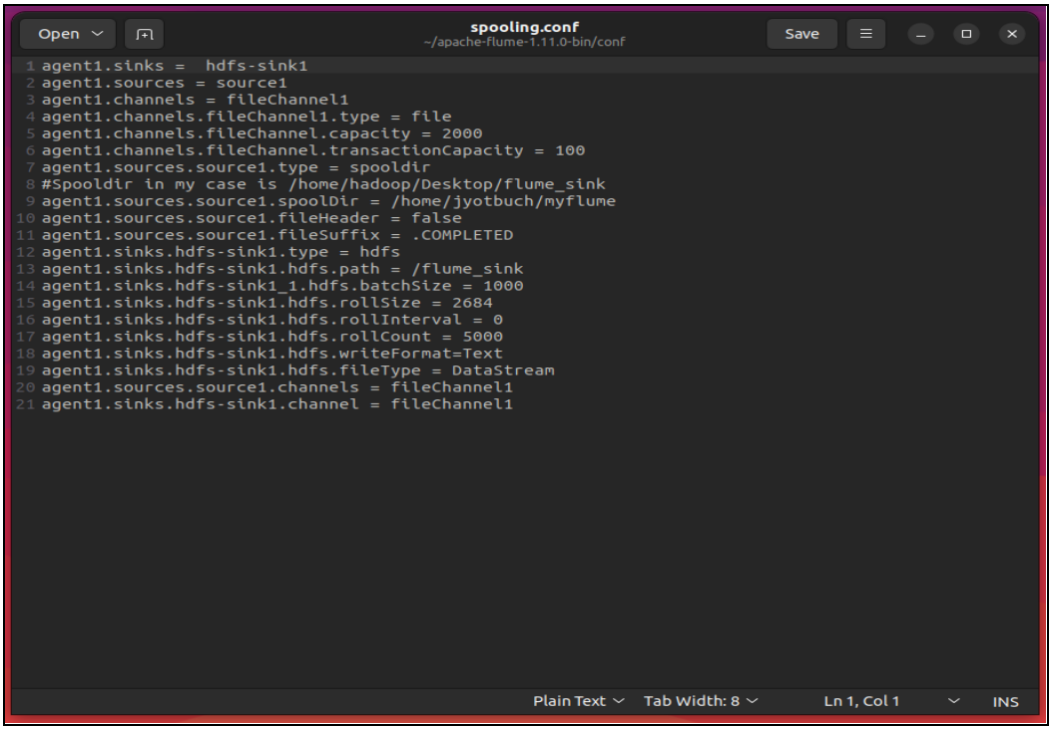
```
a1.sources= r1
a1.channels= c1 c2 c3 c4
a1.sinks= k1 k2 k3 k4

a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 4444

a1.channels.c1.type = memory
a1.channels.c1.capacity = 100

a1.channels.c2.type = memory
a1.channels.c2.capacity = 100

a1.channels.c3.type = memory
a1.channels.c3.capacity = 100
```

```
a1.channels.c4.type = memory
a1.channels.c4.capacity = 100

a1.sources.r1.interceptors = i1
a1.sources.r1.interceptors.i1.type = regex_extractor
a1.sources.r1.interceptors.i1.regex = ^(\\d)
a1.sources.r1.interceptors.i1.serializers = t
a1.sources.r1.interceptors.i1.serializers.t.name = type

a1.sources.r1.selector.type = multiplexing
a1.sources.r1.selector.header = type
a1.sources.r1.selector.mapping.1 = c1
a1.sources.r1.selector.mapping.2 = c2
a1.sources.r1.selector.mapping.3 = c3
a1.sources.r1.selector.mapping.4 = c4


a1.sinks.k1.type = hdfs
a1.sinks.k1.channel = c1
a1.sinks.k1.hdfs.path = /multi/splitIn1
a1.sinks.k1.hdfs.fileType=DataStream
a1.sinks.k1.hdfs.writeFormat=Text

a1.sinks.k2.channel = c2
a1.sinks.k2.type = hdfs
a1.sinks.k2.hdfs.path = /multi/splitIn2
a1.sinks.k2.hdfs.fileType=DataStream
a1.sinks.k2.hdfs.writeFormat=Text

a1.sinks.k3.type = hdfs
a1.sinks.k3.channel = c3
a1.sinks.k3.hdfs.path = /multi/splitIn3
a1.sinks.k3.hdfs.fileType=DataStream
a1.sinks.k3.hdfs.writeFormat=Text

a1.sinks.k4.channel = c4
a1.sinks.k4.type = hdfs
a1.sinks.k4.hdfs.path = /multi/splitIn4
a1.sinks.k4.hdfs.fileType=DataStream
```

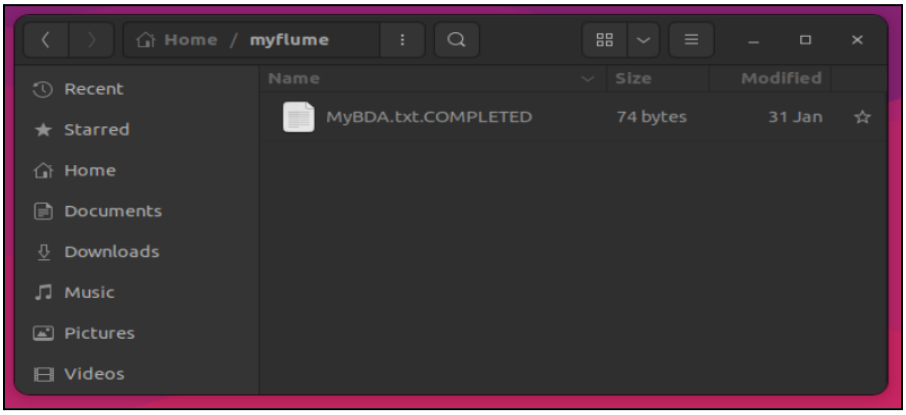
```
a1.sinks.k4.hdfs.writeFormat=Text

a1.sources.r1.channels = c1 c2 c3 c4
a1.sinks.k2.channel = c2
a1.sinks.k1.channel = c1
a1.sinks.k3.channel = c3
a1.sinks.k4.channel = c4
```

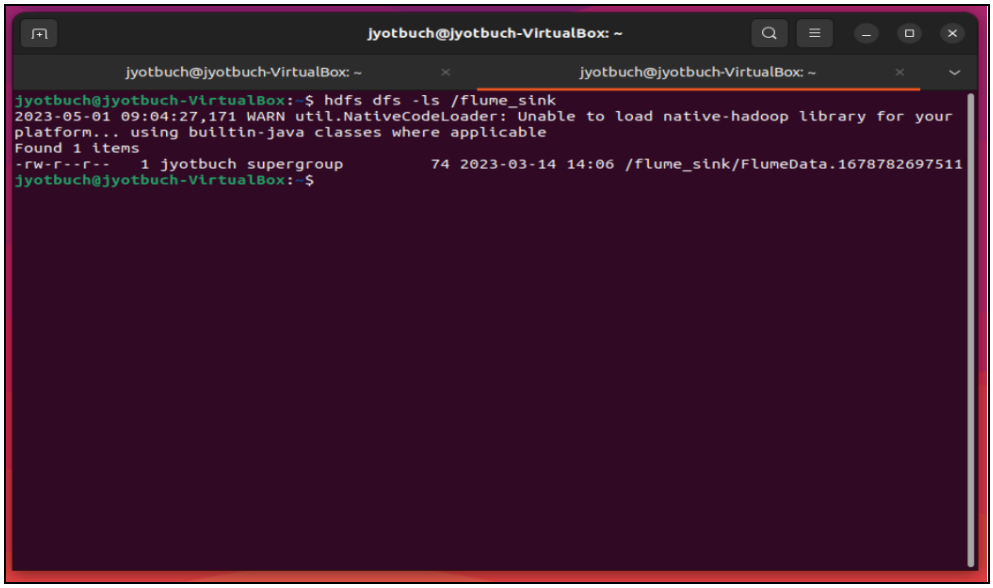


2. Apache Flume is used for data ingestion and it helps to get data from various data generators to fetch data and store it on Hadoop Distributed File System. You are required to create one text data file on Ubuntu local file system and then treat this data as source data. Use apache flume to transfer this data from source to sink of HDFS. Support your answer with screenshot of CLI fetching same text file on HDFS.



In this configuration, we define a Spooling Directory Source, a Memory Channel, and a HDFS Sink. The Spooling Directory Source monitors the specified directory (/home/user/) for new files and passes them to the Memory Channel. The HDFS Sink then writes the data to HDFS in the specified path (/user/hadoop/data/).




jyotbuch@jyotbuch-VirtualBox: ~
jyotbuch@jyotbuch-VirtualBox: ~
jyotbuch@jyotbuch-VirtualBox: ~
jyotbuch@jyotbuch-VirtualBox:~$ hdfs dfs -ls /flume_sink
2023-05-01 09:04:27,171 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 jyotbuch supergroup   74 2023-03-14 14:06 /flume_sink/FlumeData.1678782697511
jyotbuch@jyotbuch-VirtualBox:~$