Download following files from LMS link and keep them on desktop

1. book.dat
2. users.dat
3. ratings.dat

1.) Actual data is stored in mysql and flat files

A.) We need to write a sqoop job code to get the attached data loaded into HDFS and Hive tables on it by using spark HiveContext.

B.) We need to write flume agent for *filechannel and spoolDir,* to copy data from flat files in local FS to HDFS. Finally create permanent hive tables on it by using spark HiveContext

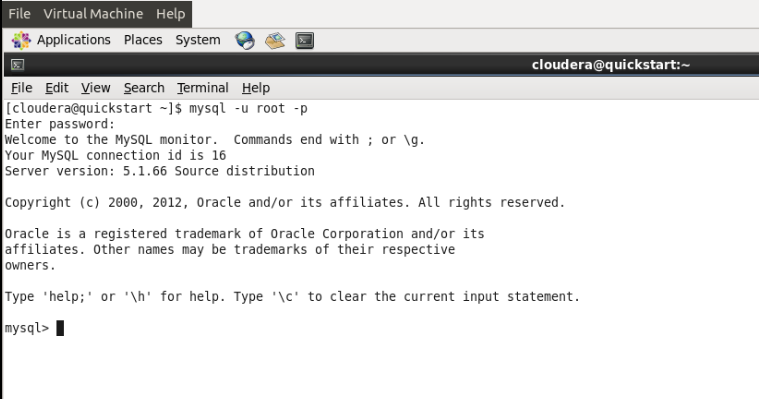
*Finally establish the connection between hive and tableau.*

MySQl:

We need to write a sqoop job code to get the attached data loaded into HDFS and Hive tables on it by using spark HiveContext.

1. Login to mysql

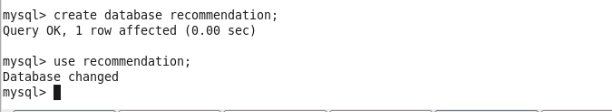
mysql -u root -p



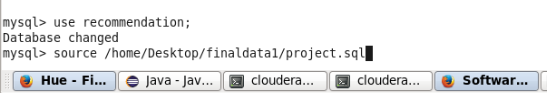
Password:cloudera

Step 1:create a database in mysql database.

Mysql> create database recommendations;



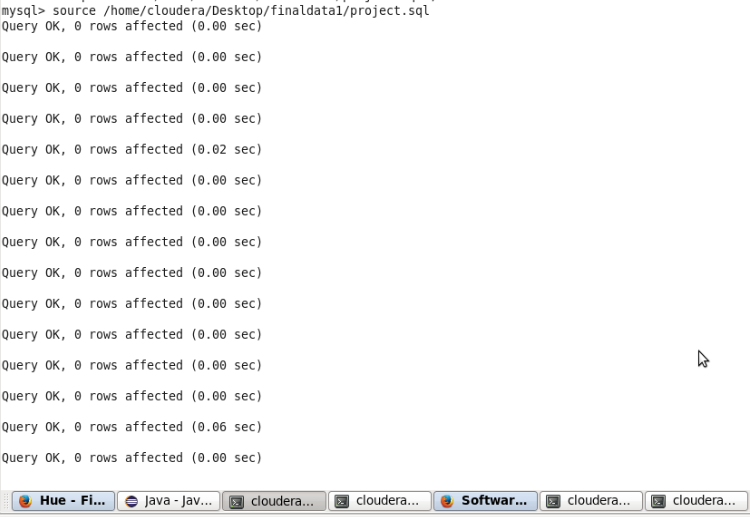
Step 2: use recommendations;



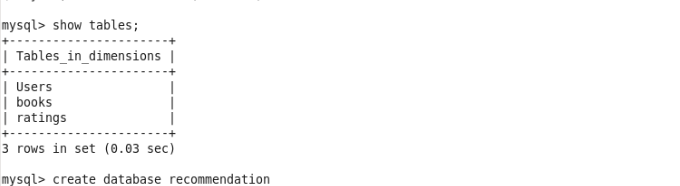
Step 3: To execute sql file for creating table and then inserting data into that table.By using source command

Mysql>source /home/Desktop/finaldata1project.sql;

// source command is used to execute sql files on mysql.



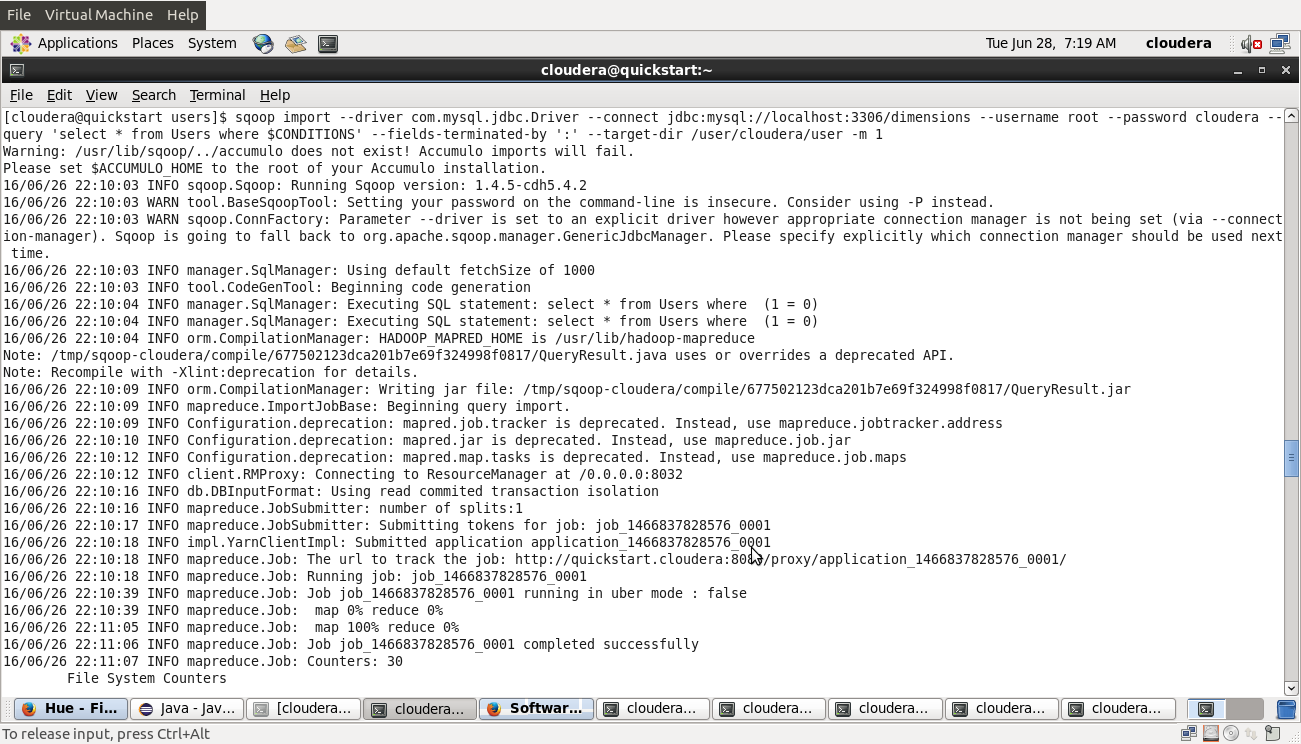
Step4: show tables;



Step 5:

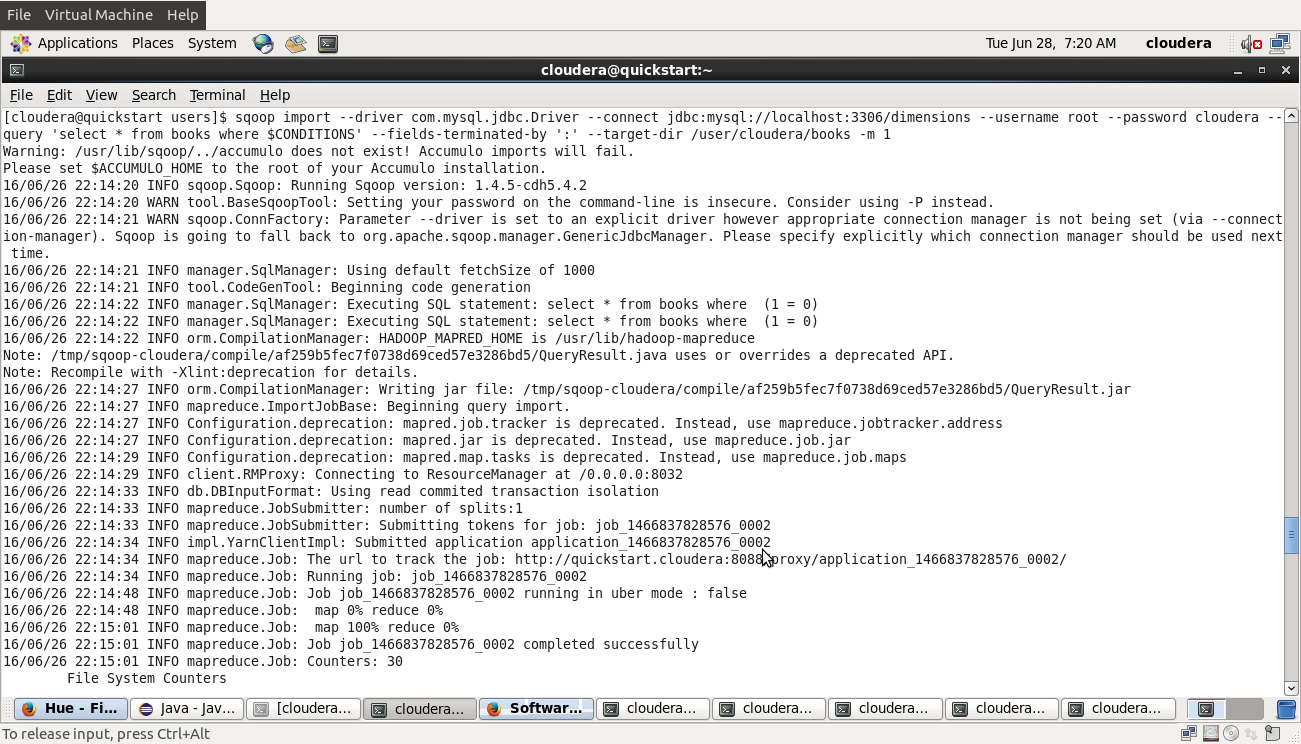
We have data on mysql and transfer data from mysql to hdfs.sqoop is a tool to provide feasible interaction between relational databases and hdfs(hadoop).

**sqoop import --driver com.mysql.jdbc.Driver --connect jdbc:mysql://localhost:3306/recommendations --username root --password hadoop --query 'select \* from Users where $CONDITIONS' --fields-terminated-by ':' --target-dir /user/cloudera/users -m 1**



Step 6:

**sqoop import --driver com.mysql.jdbc.Driver --connect jdbc:mysql://localhost:3306/recommendations --username root --password hadoop --table books --target-dir /user/cloudera/books -m 1**



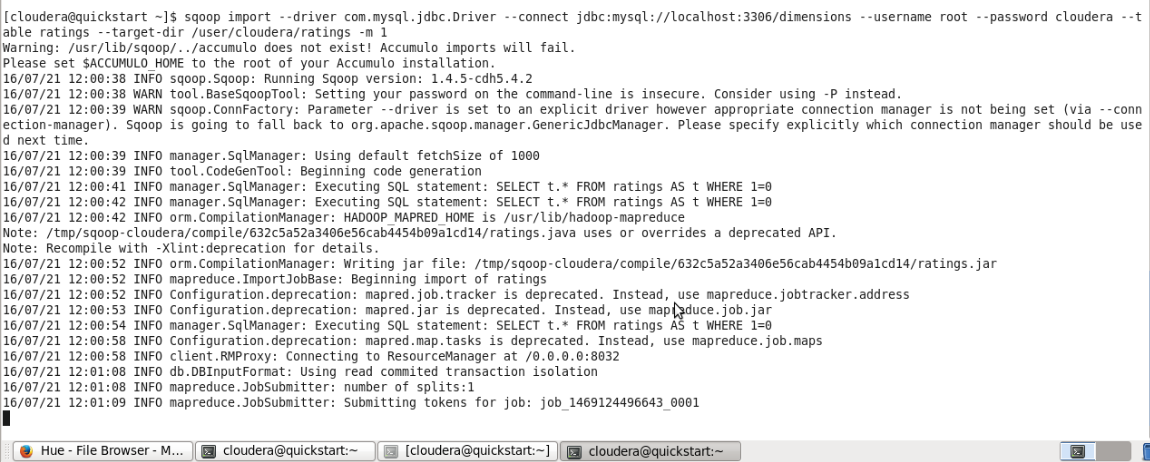
While running sqoop job,it will create a java classes and jar files.We can see that java file like this

location: local filesystem.

/tmp/sqoop-'username'/

Sqoop by default launch 4 mapper based on the boundary query tool.But we can specify user defined number of mappers by using -m 1 ------> single mapper

**sqoop import --driver com.mysql.jdbc.Driver --connect jdbc:mysql://localhost:3306/dimensions --username root --password cloudera --table ratings --target-dir /user/cloudera/ratings -m 1**



HDFS:

Mysql data stored in hdfs directory /user/cloudera/users from users data and /user/cloudera/books for books data.

**Flume:**

Ingestion data through flume:

Local FS:

Create a new folder /home/cloudera/Desktop/flumefs

mkdir /home/cloudera/Desktop/flumefs

Copy file ratings.dat into above folder

cp /home/cloudera/Desktop/finaldata1/ratings.dat /home/cloudera/flumefs/

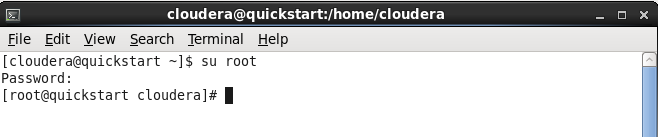
Flat file in local fs in this directory /home/cloudera/Desktop/flumefs

We need to write flume agent for filechannel and spoolDir, to copy data from flat files in local FS to HDFS. Finally create permanent hive tables on it by using spark HiveContext.

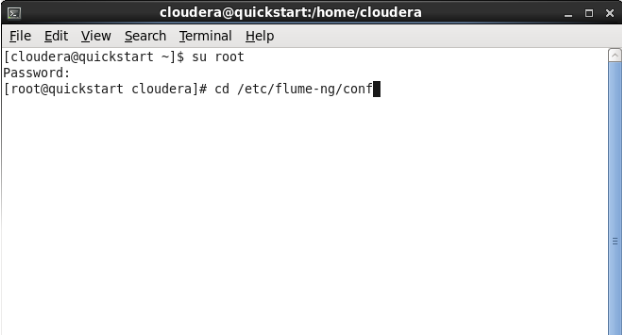
Step1: open new terminal and type

su root

Password: cloudera



cd /etc/flume-ng/conf



Type below code in agent1.conf file

vi agent1.conf

Or

gedit agent1.conf

agent1.sources.source1\_1.fileSuffix = .COMPLETED

agent1.sinks.hdfs-sink1\_1.type = hdfs

agent1.sinks.hdfs-sink1\_1.hdfs.path = hdfs://localhost:8020/user/cloudera/ratings/

agent1.sinks.hdfs-sink1\_1.hdfs.batchSize = 1000

agent1.sinks.hdfs-sink1\_1.hdfs.rollSize = 268435456

agent1.sinks.hdfs-sink1\_1.hdfs.rollInterval = 0

agent1.sinks.hdfs-sink1\_1.hdfs.rollCount = 50000000

agent1.sinks.hdfs-sink1\_1.hdfs.writeFormat=Text

agent1.sinks.hdfs-sink1\_1.hdfs.fileType = DataStream

agent1.sources.source1\_1.channels = fileChannel1\_1

agent1.sinks.hdfs-sink1\_1.channel = fileChannel1\_1

agent1.sinks = hdfs-sink1\_1

agent1.sources = source1\_1

agent1.channels = fileChannel1\_1

agent1.sources.source1\_1.spoolDir = /home/cloudera/Deskagent1.conf:

agent1.channels.fileChannel1\_1.type = file

agent1.channels.fileChannel1\_1.capacity = 200000

agent1.channels.fileChannel1\_1.transactionCapacity = 1000

agent1.sources.source1\_1.type = spooldir

agent1.sources.source1\_1.spoolDir = /home/cloudera/Desktop/flumefs/

agent1.sources.source1\_1.fileHeader = falsetop/flumefs/ratings/

spoolDir directory must contain input files i.e., ratings

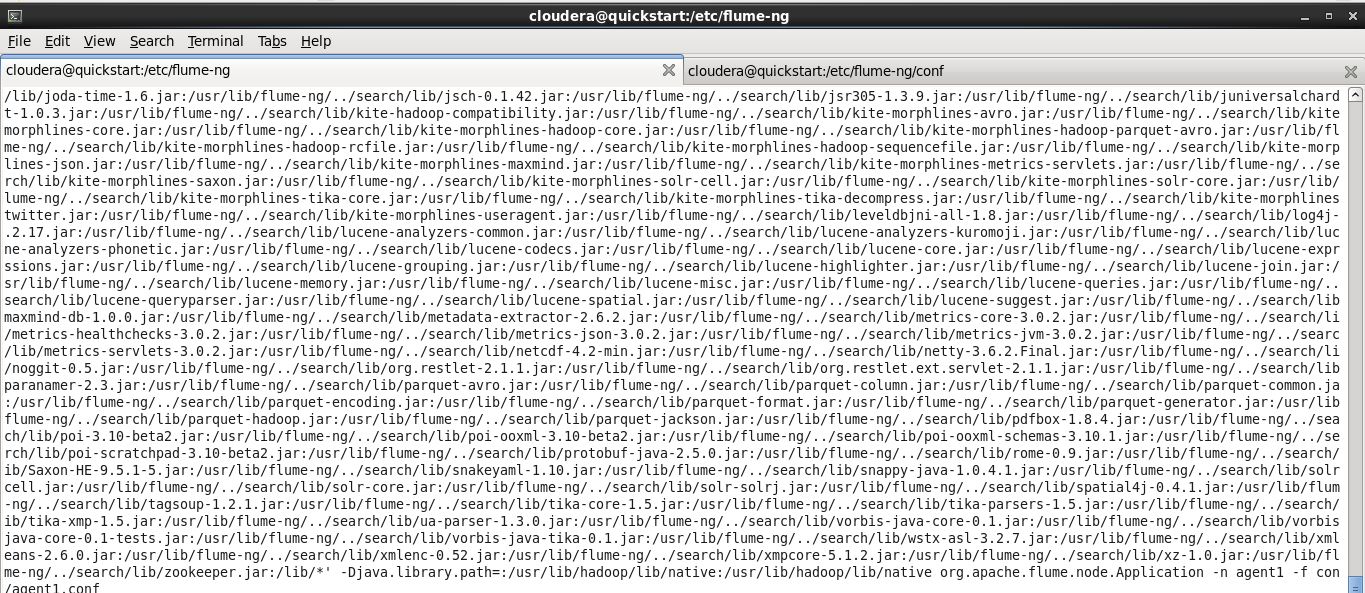


Running flume agent:

Step1:cd ..

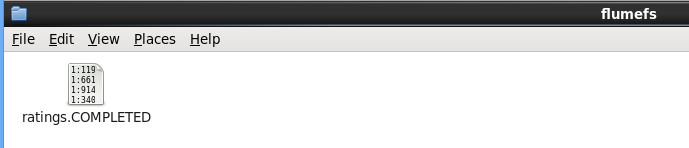
flume-ng agent -n agent1 --conf ./conf/ -f conf/agent1.conf





Go to: /home/cloudera/Desktop/flumefs

cd /home/cloudera/Desktop/flumefs



**We are store books data in hdfs through flume:**

agent1.conf:

agent1.channels.fileChannel1\_1.type = file

agent1.channels.fileChannel1\_1.capacity = 200000

agent1.channels.fileChannel1\_1.transactionCapacity = 1000

agent1.sources.source1\_1.type = spooldir

agent1.sources.source1\_1.spoolDir = /home/cloudera/Desktop/flumefs/

agent1.sources.source1\_1.fileHeader = false

agent1.sources.source1\_1.fileSuffix = .COMPLETED

agent1.sinks.hdfs-sink1\_1.type = hdfs

agent1.sinks.hdfs-sink1\_1.hdfs.path = hdfs://localhost:8020/user/cloudera/books/

agent1.sinks.hdfs-sink1\_1.hdfs.batchSize = 1000

agent1.sinks.hdfs-sink1\_1.hdfs.rollSize = 268435456

agent1.sinks.hdfs-sink1\_1.hdfs.rollInterval = 0

agent1.sinks.hdfs-sink1\_1.hdfs.rollCount = 50000000

agent1.sinks.hdfs-sink1\_1.hdfs.writeFormat=Text

agent1.sinks.hdfs-sink1\_1.hdfs.fileType = DataStream

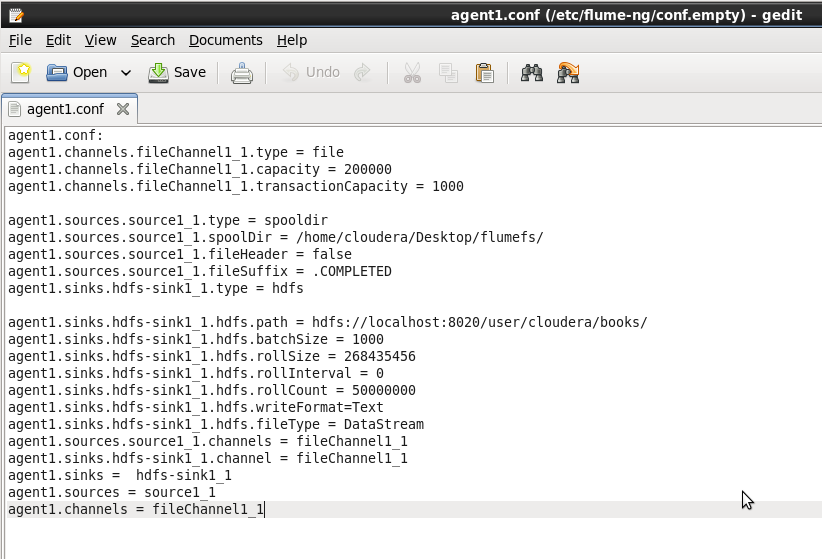
agent1.sources.source1\_1.channels = fileChannel1\_1

agent1.sinks.hdfs-sink1\_1.channel = fileChannel1\_1

agent1.sinks = hdfs-sink1\_1

agent1.sources = source1\_1

agent1.channels = fileChannel1\_1

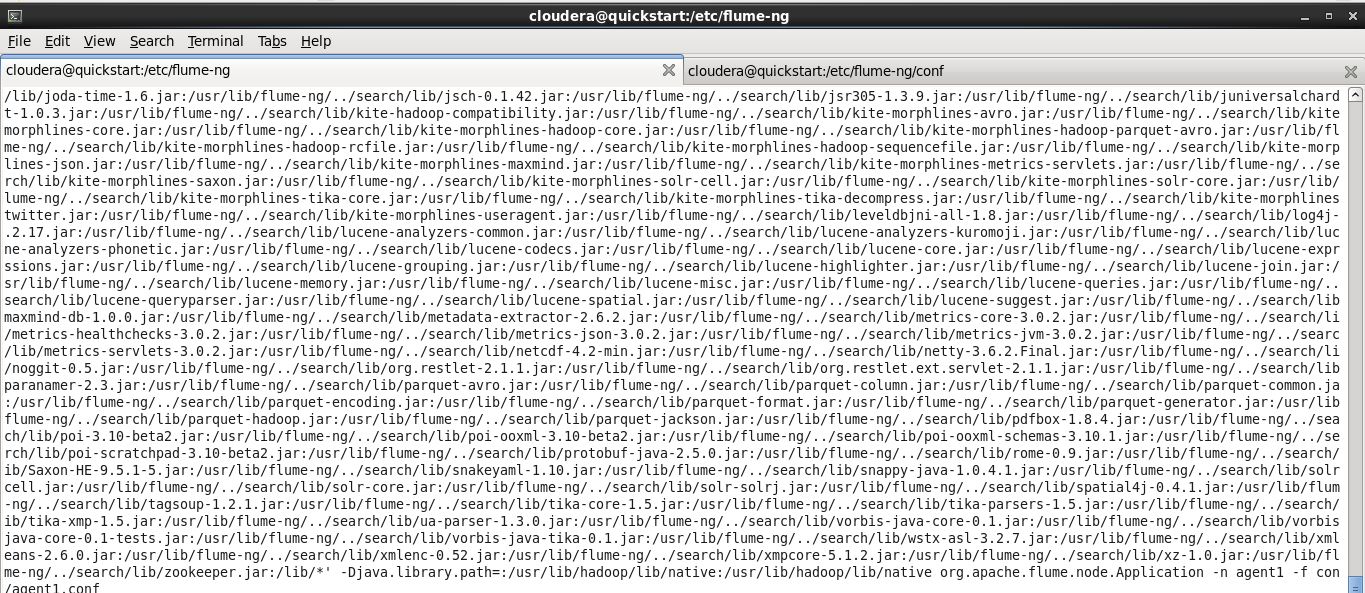


Running flume agent:

Step1:**cd ..**

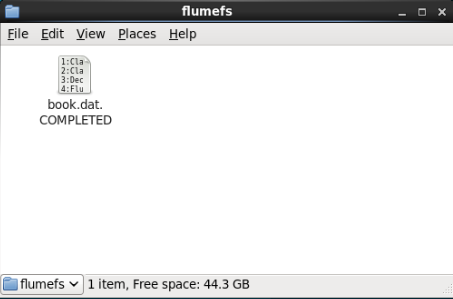
**flume-ng agent -n agent1 --conf ./conf/ -f conf/agent1.conf**





Go to: **/home/cloudera/Desktop/flumefs**

**cd /home/cloudera/Desktop/flumefs**



**We are store users data in hdfs through flume:**

agent1.conf:

agent1.channels.fileChannel1\_1.type = file

agent1.channels.fileChannel1\_1.capacity = 200000

agent1.channels.fileChannel1\_1.transactionCapacity = 1000

agent1.sources.source1\_1.type = spooldir

agent1.sources.source1\_1.spoolDir = /home/cloudera/Desktop/flumefs1/

agent1.sources.source1\_1.fileHeader = false

agent1.sources.source1\_1.fileSuffix = .COMPLETED

agent1.sinks.hdfs-sink1\_1.type = hdfs

agent1.sinks.hdfs-sink1\_1.hdfs.path = hdfs://localhost:8020/user/cloudera/books/

agent1.sinks.hdfs-sink1\_1.hdfs.batchSize = 1000

agent1.sinks.hdfs-sink1\_1.hdfs.rollSize = 268435456

agent1.sinks.hdfs-sink1\_1.hdfs.rollInterval = 0

agent1.sinks.hdfs-sink1\_1.hdfs.rollCount = 50000000

agent1.sinks.hdfs-sink1\_1.hdfs.writeFormat=Text

agent1.sinks.hdfs-sink1\_1.hdfs.fileType = DataStream

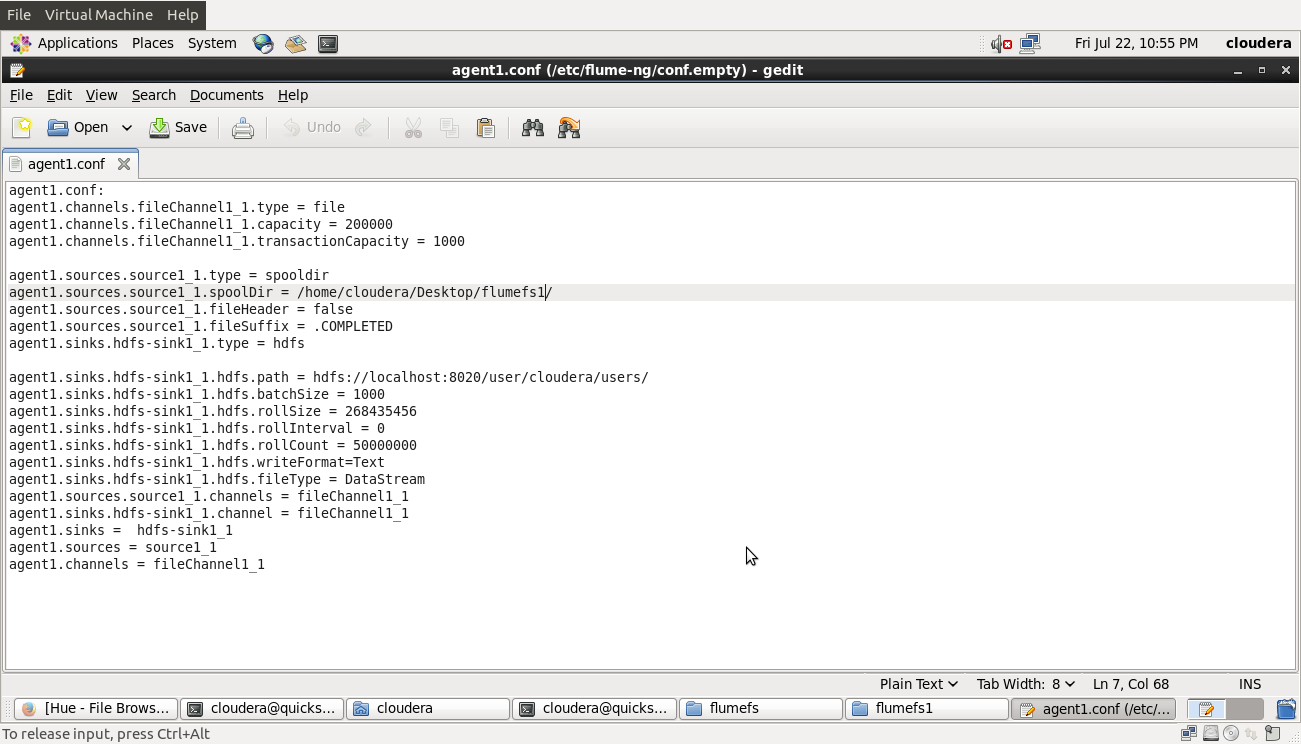
agent1.sources.source1\_1.channels = fileChannel1\_1

agent1.sinks.hdfs-sink1\_1.channel = fileChannel1\_1

agent1.sinks = hdfs-sink1\_1

agent1.sources = source1\_1

agent1.channels = fileChannel1\_1

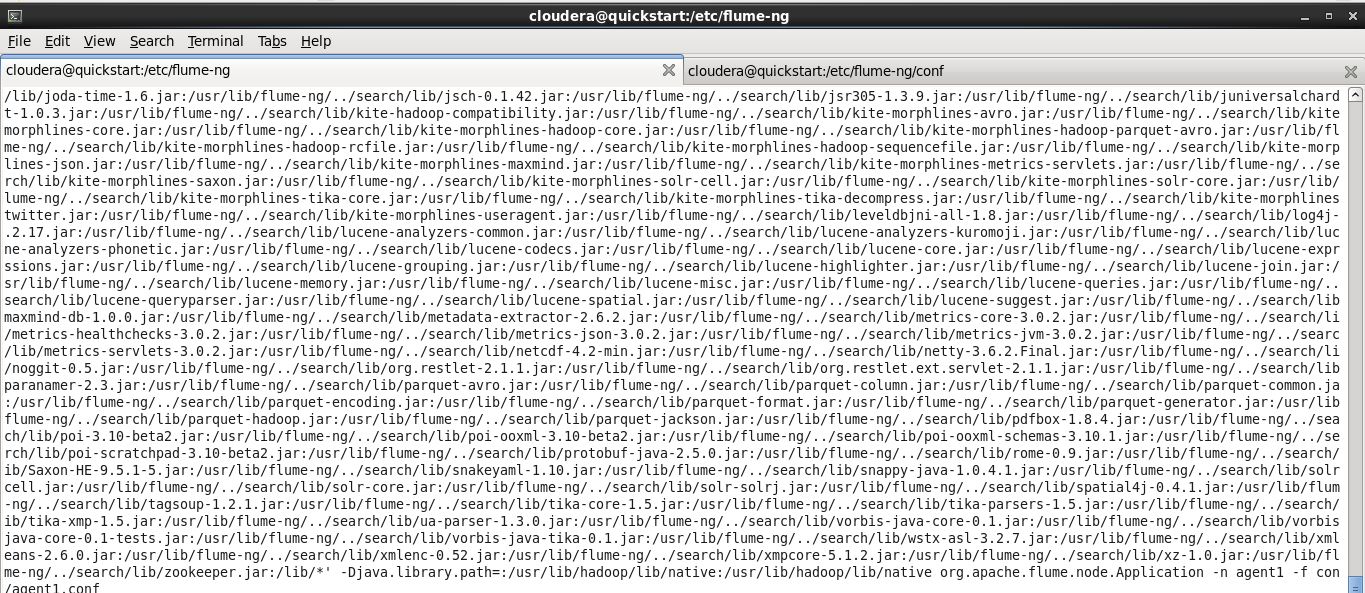


Running flume agent:

Step1:**cd ..**

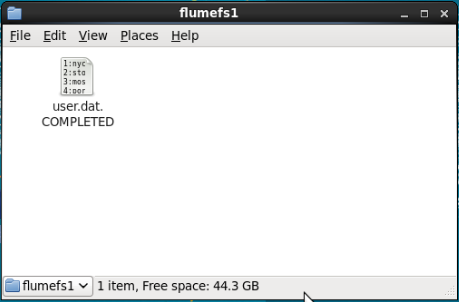
**flume-ng agent -n agent1 --conf ./conf/ -f conf/agent1.conf**





Go to: **/home/cloudera/Desktop/flumefs1**

**cd /home/cloudera/Desktop/flumefs1**



**All three file store in HDFS**

