1. Create directory **/flume\_import** on hdfs.

**hadoop fs -mkdir /flume\_import**

above command will create flume\_import directory on hdfs.

mkdir: to create new directory.



1. Copy dataset from local file system to HDFS using flume :

Run the below flume command to copy the dataset from local filesystem to HDFS

**flume-ng agent -n agent1 -c conf -f /home/acadgild/flume/conf/filecopy.conf**

Command Description:

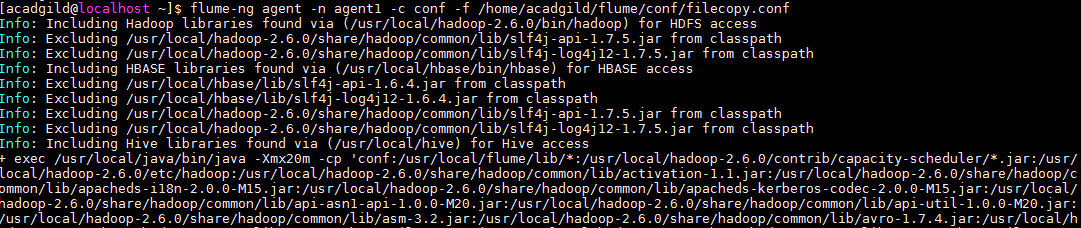
**flume-ng: to start the flume agent**

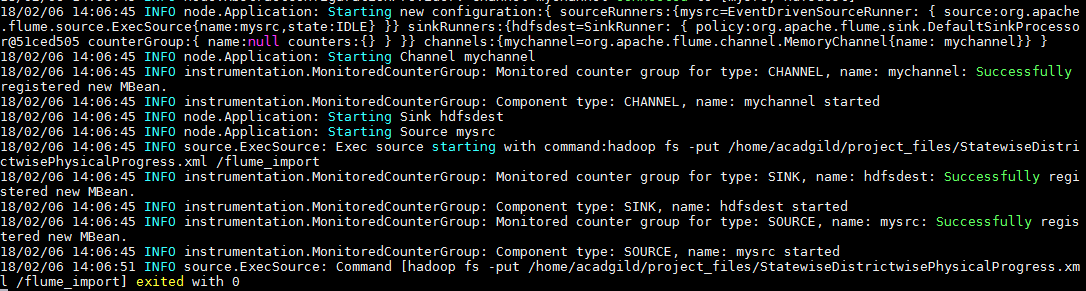
**agent -n agent1 : to provide the agent name.**

**-c conf -f : Configuration file path.**

Attached is the dataset and flume configuration file:

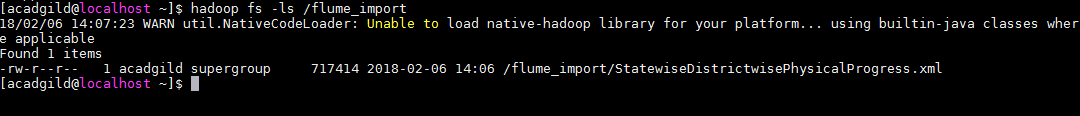
** **





Now check the file on HDFS using below command:

**hadoop fs -ls /flume\_import**



Problem statement

1. Find out the districts who achieved 100 percent objective in BPL cards.

**DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();**

**XML\_DATA = LOAD '/flume\_import/StatewiseDistrictwisePhysicalProgress.xml' using org.apache.pig.piggybank.storage.XMLLoader('row') as (x:chararray);**

**parsed\_data = foreach XML\_DATA generate XPath(x,'row/District\_Name'),XPath(x,'row/Project\_Objectives\_IHHL\_BPL'),XPath(x,'row/Project\_Performance-IHHL\_BPL');**

**result = FILTER parsed\_data by ($1==$2);**

**store result into '/flume\_export' using PigStorage(',');**

Commands Description :

**DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();**

As we need to parse the xml data so we are using the above command

**DEFINE XPath:** This command will defile XPath and will use the class org.apache.pig.piggybank.evaluation.xml.XPath();

**org.apache.pig.piggybank.evaluation.xml.XPath() :** XPath function for parsing xml files.

**XML\_DATA = LOAD '/flume\_import/StatewiseDistrictwisePhysicalProgress.xml' using org.apache.pig.piggybank.storage.XMLLoader('row') as (x:chararray);**

Parses an XML input file given a specified identifier of tags to be loaded. The output is a bag of XML elements where each element is returned as a chararray containing the text of the matched XML element including the start and tags as well as the data between them. In case of nesting elements of the matching tags, only the top level one is returned.

**parsed\_data = foreach XML\_DATA generate XPath(x,'row/District\_Name'),XPath(x,'row/Project\_Objectives\_IHHL\_BPL'),XPath(x,'row/Project\_Performance-IHHL\_BPL');**

Now we are using XPath to filter out the required columns (District\_Name, Project\_Objectives\_IHHL\_BPL , Project\_Performance-IHHL\_BPL') that will be used further.

**result = FILTER parsed\_data by ($1==$2);**

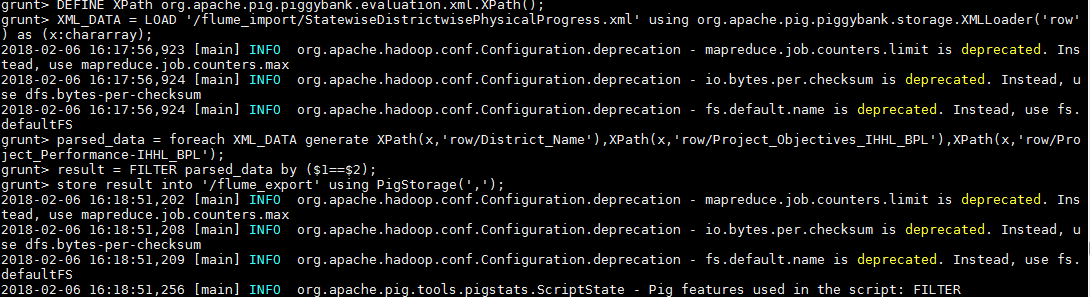
we will use FILTER command to filter the data where

Project\_Objectives\_IHHL\_BPL= Project\_Performance-IHHL\_BPL

It will indicate 100% objective in BPL cards.

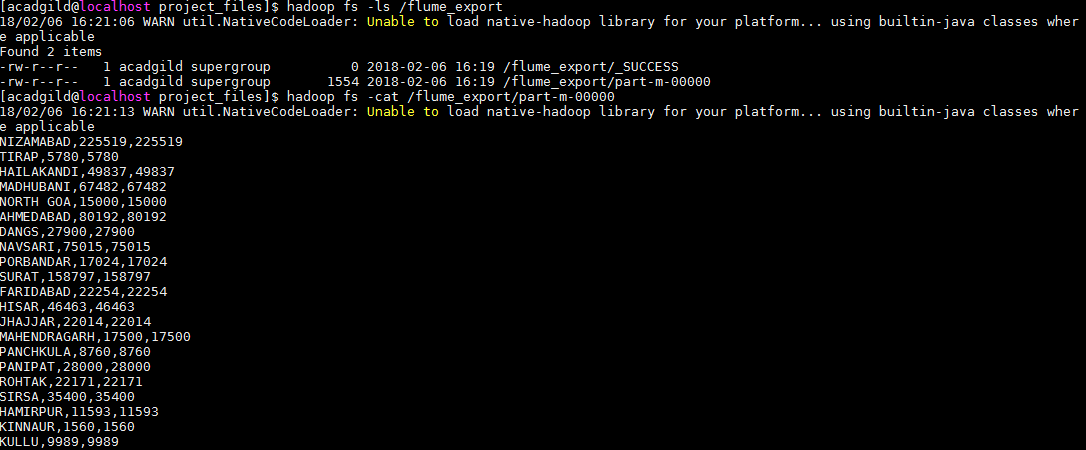
**store result into '/flume\_export' using PigStorage(',');**

store command will store the data into the provided hdfs directory using delimiter as ‘,’.



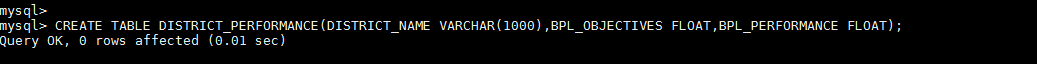
Check the data inside the hdfs directory /flume\_export.

**hadoop fs -cat /flume\_export/part-m-00000**



Now we will create table in mysql in which data will be loaded from HDFS using sqoop.

**CREATE TABLE DISTRICT\_PERFORMANCE(DISTRICT\_NAME VARCHAR(1000),BPL\_OBJECTIVES FLOAT,BPL\_PERFORMANCE FLOAT);**



Run the below sqoop export command to export the data from HDFS to MySQL :

**sqoop export --connect jdbc:mysql://localhost/mydatabase --username root --table DISTRICT\_PERFORMANCE --export-dir /flume\_export --m 1**

Command Description :

sqoop export : it will export the data from hdfs to rdbms

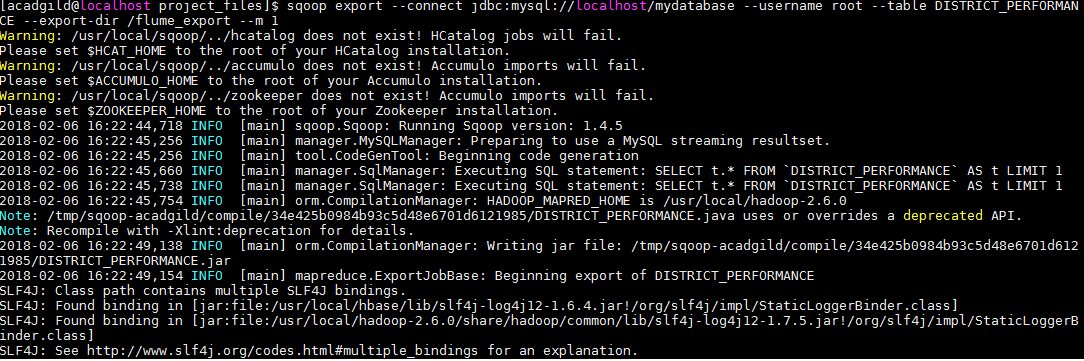
--connect jdbc:mysql://localhost/mydatabase : connection string for MySQL RDBMS for database “mydatabase”

--username: username parameter

--table: tablename parameter

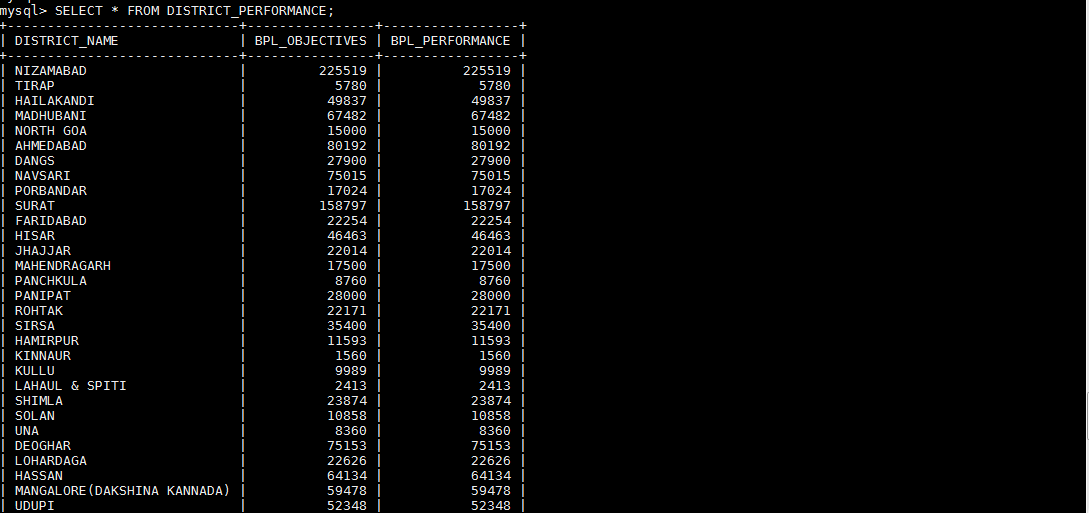
--export-dir /flume\_export : directory from where data will be exported.

--m : number of mappers



Finally check the data in MySQL table:

**SELECT \* FROM DISTRICT\_PERFORMANCE;**



1. Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards.

Create pig UDF in java and export is as Jar, here we have created Pig UDF by name “FilterBplCards”

Add the jar to Pig using below command :

**register '/home/acadgild/project\_files/BplCards\_udf.jar';**

Define the UDF using below command :

**define FilterBplCards FilterBplCards();**

**DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();**

**XML\_DATA = LOAD '/flume\_import/StatewiseDistrictwisePhysicalProgress.xml' using org.apache.pig.piggybank.storage.XMLLoader('row') as (x:chararray);**

**parsed\_data = foreach XML\_DATA generate XPath(x,'row/District\_Name'),XPath(x,'row/Project\_Objectives\_IHHL\_BPL'),XPath(x,'row/Project\_Performance-IHHL\_BPL');**

**Bpl\_Data = foreach parsed\_data generate FilterBplCards(CONCAT($0,',',$1,',',$2));**

**result = FILTER Bpl\_Data by ($0!='');**

**store result into '/Bpl\_80percent';**

**Commands Description:**

**register '/home/acadgild/project\_files/BplCards\_udf.jar'**

it will register the jar that we have created for udf.

**define FilterBplCards FilterBplCards()**

define udf class with function so that it can be used in pig.

**DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();**

As we need to parse the xml data so we are using the above command

**DEFINE XPath:** This command will defile XPath and will use the class org.apache.pig.piggybank.evaluation.xml.XPath();

**org.apache.pig.piggybank.evaluation.xml.XPath() :** XPath function for parsing xml files.

**XML\_DATA = LOAD '/flume\_import/StatewiseDistrictwisePhysicalProgress.xml' using org.apache.pig.piggybank.storage.XMLLoader('row') as (x:chararray);**

Parses an XML input file given a specified identifier of tags to be loaded. The output is a bag of XML elements where each element is returned as a chararray containing the text of the matched XML element including the start and tags as well as the data between them. In case of nesting elements of the matching tags, only the top level one is returned.

**parsed\_data = foreach XML\_DATA generate XPath(x,'row/District\_Name'),XPath(x,'row/Project\_Objectives\_IHHL\_BPL'),XPath(x,'row/Project\_Performance-IHHL\_BPL');**

Now we are using XPath to filter out the required columns (District\_Name, Project\_Objectives\_IHHL\_BPL , Project\_Performance-IHHL\_BPL') that will be used further.

**foreach parsed\_data generate FilterBplCards(CONCAT($0,',',$1,',',$2));**

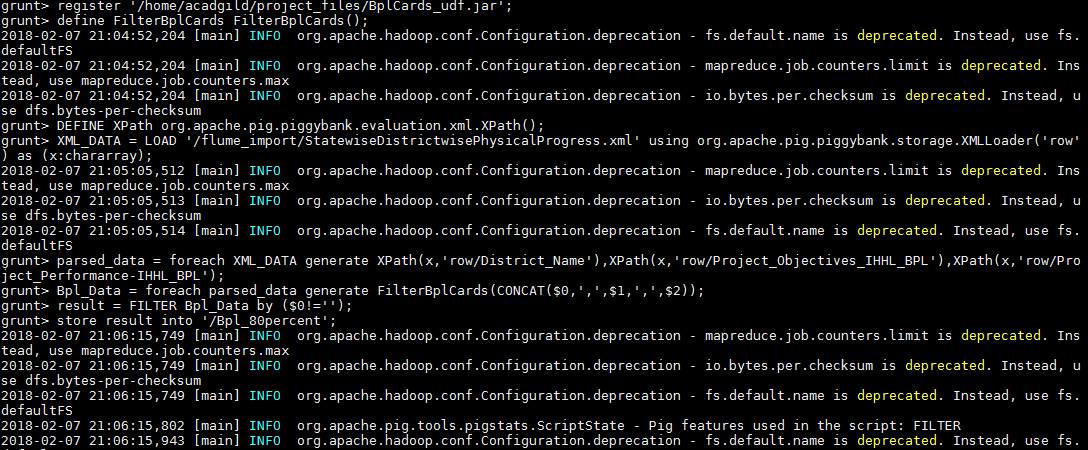
calling the FilterBplCards udf for each tuple of parsed\_data relation.

**FILTER Bpl\_Data by ($0!='');**

Filtering out blank tuples.

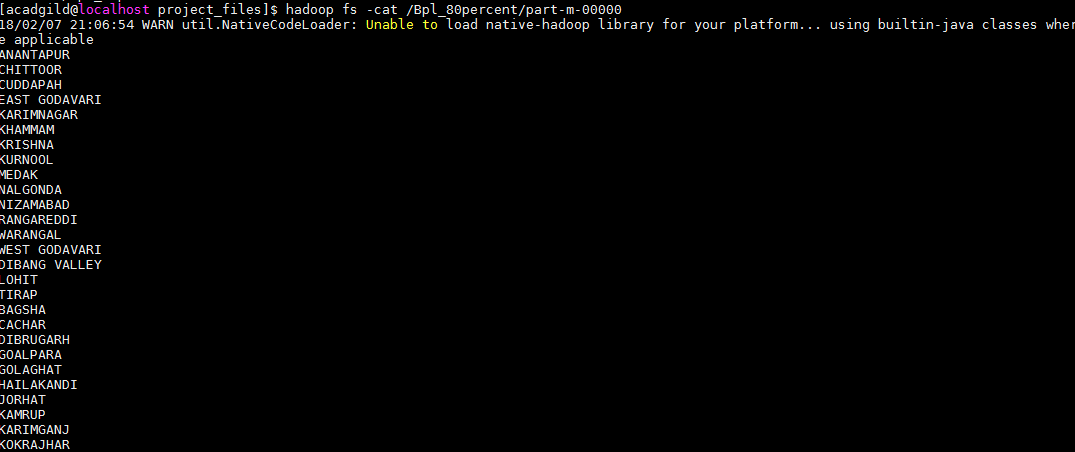
**store result into '/Bpl\_80percent';**

store command will store the data into the provided hdfs directory



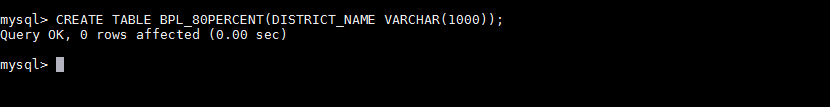
Check the data in the directory /Bpl\_80percent using below command:

**hadoop fs -cat /Bpl\_80percent/part-m-00000**



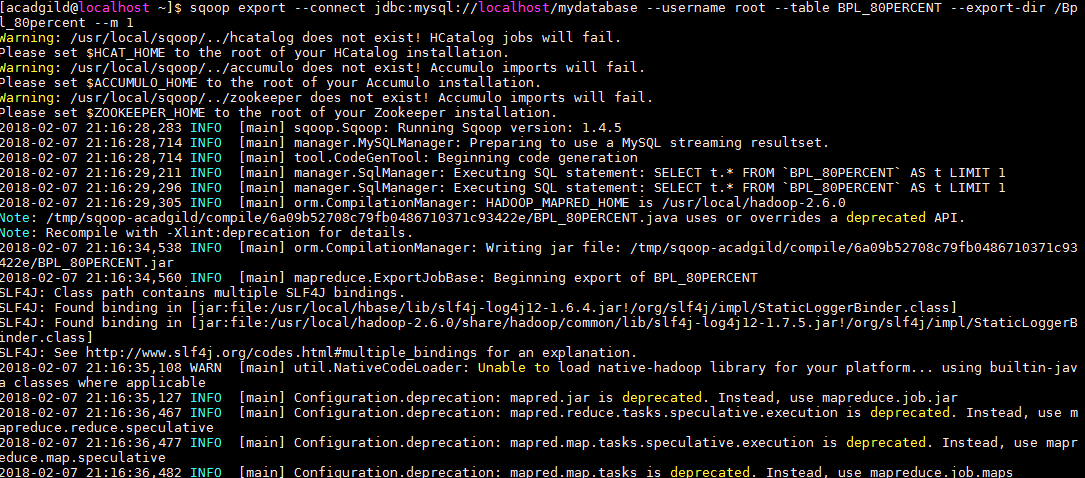
Now we will create table in mysql in which data will be loaded from HDFS using sqoop.

**CREATE TABLE BPL\_80PERCENT(DISTRICT\_NAME VARCHAR(1000));**



Run the below sqoop export command to export the data from HDFS to MySQL :

**sqoop export --connect jdbc:mysql://localhost/mydatabase --username root --table BPL\_80PERCENT --export-dir /Bpl\_80percent --m 1**



Finally check the data in MySQL table:

**SELECT \* FROM BPL\_80PERCENT;**

