**Assignment 2**

**30%** due on 23:59 Sunday Feb 24 2019

Hadoop Data Ingestion

Apahe Sqoop is a tool used to extract data from structured data sources into Hadoop. It can be used to transfer huge data between Hadoop and other relational database systems. We can import the data from Relational databases such as MySQL, Oracle into Hadoop or HBase for processing purpose and after completing processing we can again export the data back into RDBMS. The processing can be done with Map-Reduce programs or Hive.

Apache Flume acts as a system used to write data to Apache Hadoop and Apache HBase in a reliable and scalable fashion and we can write data into it in any format and language such as MapReduce, Hive, Pig, and Impala.

In this assignment, we are going to ingest data from different sources into different destinations.

To start, run your Cloudera virtual machine. You will use Linux command-line to interact with Sqoop, Flume and CentOS operating system.

You are highly recommended to read the online documentation while doing this assignment:

https://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html

https://flume.apache.org/FlumeUserGuide.html

<http://hbase.apache.org/0.94/book/ops_mgt.html>

**Provided files:**

The employees sample database, installed while working the last workshop.

**Sqoop Data import 50%**

You will use Apache Sqoop to import data from MySQL server into Apache Hive and Apache HBase.

**Question 1:**

Write a Sqoop command to import the **employees** table from MySQL employees database into a Hive partition. This partition should contains only rows with ‘gender’ = ‘F’ (female) from the original table. This partition belongs to an existing Hive managed table employees\_part in the emp\_mysql database.

*Note: (employees\_part table was created while working on the workshop and already contains a Hive partition with ‘gender’=’M’)*

sqoop import --connect jdbc:mysql://localhost/employees --username cloudera --password cloudera --query 'select emp\_no,birth\_date,first\_name,last\_name,hire\_date from employees where gender="F" AND $CONDITIONS' --direct -m 1 --split-by emp\_no --hive-import --create-hive-table --hive-table emp\_mysql.employees\_part --hive-partition-key gender --hive-partition-value 'F' --target-dir '/user/hive/warehouse/emp\_mysql.db/employees\_part/gender=F'

**Question 2:**

Write a HiveQL script to count the rows in each partition of employees\_part

use emp\_mysql;

select count(\*) from employees\_part where gender = 'M';

select count(\*) from employees\_part where gender = 'F';

**Question 3:**

Write a Sqoop command to import the employees table from MySQL employees database into HBase.

The Sqoop command should create a new HBase table employees\_hbase, column family infos and use emp\_no as rowkey

sqoop import --connect jdbc:mysql://localhost/employees --username cloudera --password cloudera --table employees --hbase-table employees\_hbase --hbase-row-key emp\_no --column-family infos --hbase-create-table -m 8

**Flume Data ingestion 50%**

Fan out is the process of delivering events form one source to multiple sinks through multiple channels. We have 2 modes for fan out, they are replicating and multiplexing. In the replicating flow, the event is sent to all the configured channels. In the multiplexing flow, the event is sent to only a subset of channels.

**Question 1:**

The data source is a financial stream. We want to send data to a particular HDFS directory based on the currency symbol (EUR, USD).

You will simulate the data source using telnet by typing the currency symbol manually.

Write a Flume agent configuration file that:

* Collect data from a netcat source
* Event with ‘EUR‘ symbol is routed to the ‘EUR’ directory on HDSF
* Event with ‘USD’ symbol is routed to the ‘USD’ directory on HDFS
* All other events are routed to the ‘GNL’ directory on HDFS

agt1.sources = src1

agt1.sinks = HDFS\_EUR HDFS\_USD HDFS\_GNL

agt1.channels = MemChannel\_EUR MemChannel\_USD MemChannel\_GNL

# Describe/configure the source

agt1.sources.src1.type = netcat

agt1.sources.src1.bind = localhost

agt1.sources.src1.port = 45454

agt1.sources.src1.interceptors = i1

agt1.sources.src1.interceptors.i1.type = regex\_extractor

agt1.sources.src1.interceptors.i1.regex = (EUR|USD)

agt1.sources.src1.interceptors.i1.serializers = s1

agt1.sources.src1.interceptors.i1.serializers.s1.name = BigData

agt1.sources.src1.selector.type = multiplexing

agt1.sources.src1.selector.header = BigData

agt1.sources.src1.selector.mapping.EUR = MemChannel\_EUR

agt1.sources.src1.selector.mapping.USD = MemChannel\_USD

agt1.sources.src1.selector.default = MemChannel\_GNL

# Bind the source and sink to the channel

agt1.sources.src1.channels = MemChannel\_EUR MemChannel\_USD MemChannel\_GNL

# Use a channel which buffers events in memory

agt1.channels.MemChannel\_EUR.type = memory

agt1.channels.MemChannel\_EUR.capacity = 1000

agt1.channels.MemChannel\_EUR.transactionCapacity = 100

agt1.channels.MemChannel\_USD.type = memory

agt1.channels.MemChannel\_USD.capacity = 1000

agt1.channels.MemChannel\_USD.transactionCapacity = 100

agt1.channels.MemChannel\_GNL.type = memory

agt1.channels.MemChannel\_GNL.capacity = 1000

agt1.channels.MemChannel\_GNL.transactionCapacity = 100

agt1.sinks.HDFS\_EUR.channel = MemChannel\_EUR

agt1.sinks.HDFS\_EUR.type = hdfs

agt1.sinks.HDFS\_EUR.hdfs.path = /user/flume/EUR

agt1.sinks.HDFS\_EUR.hdfs.fileType = DataStream

agt1.sinks.HDFS\_EUR.hdfs.writeFormat = Text

agt1.sinks.HDFS\_EUR.hdfs.batchSize = 1000

agt1.sinks.HDFS\_EUR.hdfs.rollSize = 0

agt1.sinks.HDFS\_EUR.hdfs.rollCount = 100000

agt1.sinks.HDFS\_USD.channel = MemChannel\_USD

agt1.sinks.HDFS\_USD.type = hdfs

agt1.sinks.HDFS\_USD.hdfs.path = /user/flume/USD

agt1.sinks.HDFS\_USD.hdfs.fileType = DataStream

agt1.sinks.HDFS\_USD.hdfs.writeFormat = Text

agt1.sinks.HDFS\_USD.hdfs.batchSize = 1000

agt1.sinks.HDFS\_USD.hdfs.rollSize = 0

agt1.sinks.HDFS\_USD.hdfs.rollCount = 100000

agt1.sinks.HDFS\_GNL.channel = MemChannel\_GNL

agt1.sinks.HDFS\_GNL.type = hdfs

agt1.sinks.HDFS\_GNL.hdfs.path = /user/flume/GNL

agt1.sinks.HDFS\_GNL.hdfs.fileType = DataStream

agt1.sinks.HDFS\_GNL.hdfs.writeFormat = Text

agt1.sinks.HDFS\_GNL.hdfs.batchSize = 1000

agt1.sinks.HDFS\_GNL.hdfs.rollSize = 0

agt1.sinks.HDFS\_GNL.hdfs.rollCount = 100000

**Question 2:**

Write the Flume command to run the agent

flume-ng agent -n agt1 -f P2Q1.conf -c conf