

Kalyan Big Data Projects – Project 11

How To Stream CSV Data Into Hbase Using Apache Flume

Pre-Requisites of Flume Project:

hadoop-2.6.0
flume-1.6.0
hbase-0.98.4
java-1.7

Project Compatibility :

1. hadoop-2.6.0 + hbase-0.98.4 + flume-1.6.0
2. hadoop-2.7.2 + hbase-1.1.2 + flume-1.7.0

NOTE: Make sure that install all the above components

Flume Project Download Links:

`hadoop-2.6.0.tar.gz` ==> [link](https://archive.apache.org/dist/hadoop/core/hadoop-2.6.0/hadoop-2.6.0.tar.gz)
(<https://archive.apache.org/dist/hadoop/core/hadoop-2.6.0/hadoop-2.6.0.tar.gz>)

`apache-flume-1.6.0-bin.tar.gz` ==> [link](https://archive.apache.org/dist/flume/1.6.0/apache-flume-1.6.0-bin.tar.gz)
(<https://archive.apache.org/dist/flume/1.6.0/apache-flume-1.6.0-bin.tar.gz>)

`hbase-1.1.2-bin.tar.gz` ==> [link](https://archive.apache.org/dist/hbase/1.1.2/hbase-1.1.2-bin.tar.gz)
(<https://archive.apache.org/dist/hbase/1.1.2/hbase-1.1.2-bin.tar.gz>)

`phoenix-4.7.0-HBase-1.1-bin.tar.gz` ==> [link](https://archive.apache.org/dist/phoenix/phoenix-4.7.0-HBase-1.1/bin/phoenix-4.7.0-HBase-1.1-bin.tar.gz)
(<https://archive.apache.org/dist/phoenix/phoenix-4.7.0-HBase-1.1/bin/phoenix-4.7.0-HBase-1.1-bin.tar.gz>)

`kalyan-bigdata-examples.jar` ==> [link](https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/kalyan/kalyan-bigdata-examples.jar)
(<https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/kalyan/kalyan-bigdata-examples.jar>)

`kalyan-flume-project-0.1.jar` ==> [link](https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/kalyan/kalyan-flume-project-0.1.jar)
(<https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/kalyan/kalyan-flume-project-0.1.jar>)

`kalyan-csv-hbase-agent.conf` ==> [link](https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/flume/project11-hbase-csv/kalyan-csv-hbase-agent.conf)
(<https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/flume/project11-hbase-csv/kalyan-csv-hbase-agent.conf>)

Learnings of this Project:

- We will learn Flume Configurations and Commands
 - Flume Agent
 1. Source (Exec Source)
 2. Channel (Memory Channel)
 3. Sink (Hbase Sink)
 - Major project in Real Time `Product Log Analysis`
 1. We are extracting the data from server logs
 2. This data will be useful to do analysis on product views
 3. CSV is the output format
 - We can use hbase to analyze this data
-

1. create "**kalyan-csv-hbase-agent.conf**" file with below content

```
agent.sources = EXEC
agent.channels = MemChannel
agent.sinks = HBASE
```

```
agent.sources.EXEC.type = exec
agent.sources.EXEC.command = tail -F /tmp/users.csv
agent.sources.EXEC.channels = MemChannel
```

```
agent.sinks.HBASE.type = hbase
agent.sinks.HBASE.table = users3
agent.sinks.HBASE.columnFamily = cf
agent.sinks.HBASE.serializer = com.orienit.kalyan.flume.sink.CsvHbaseEventSerializer
agent.sinks.HBASE.serializer.delimiter = ,
agent.sinks.HBASE.serializer.colNames = userid,username,password,email,country,state,city,dt
agent.sinks.HBASE.channel = MemChannel
```

```
agent.channels.MemChannel.type = memory
agent.channels.MemChannel.capacity = 1000
agent.channels.MemChannel.transactionCapacity = 100
```

2. Copy "**kalyan-csv-hbase-agent.conf**" file into "**\$FUME_HOME/conf**" folder

3. Copy "**kalyan-flume-project-0.1.jar** and **kalyan-bigdata-examples.jar**" files into "**\$FLUME_HOME/lib**" folder

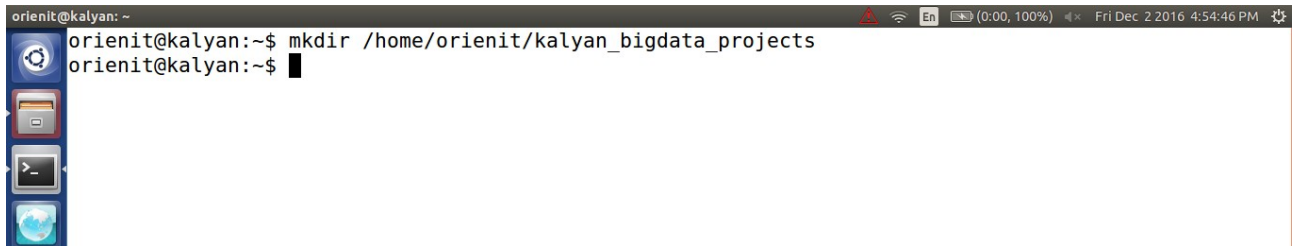
4. Generate Large Amount of Sample JSON data follow this [article](#).

(<http://kalyanbigdatatraining.blogspot.com/2016/12/how-to-generate-large-amount-of-sample.html>)

5. Follow below steps...

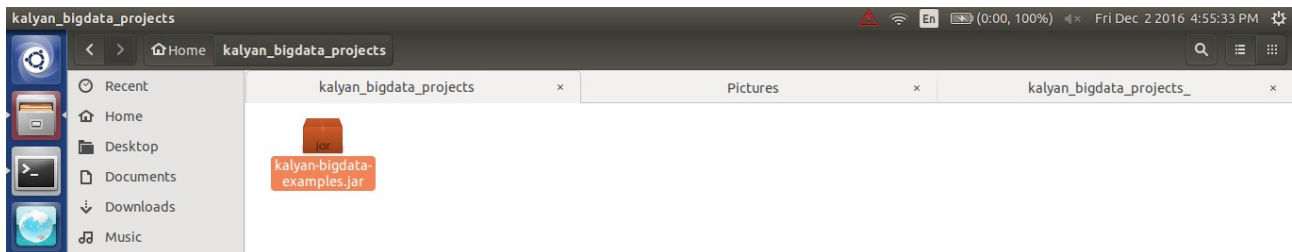
i) Create '**kalyan_bigdata_projects**' folder in **user home** (i.e **/home/orienit**)

Command: `mkdir /home/orienit/kalyan_bigdata_projects`



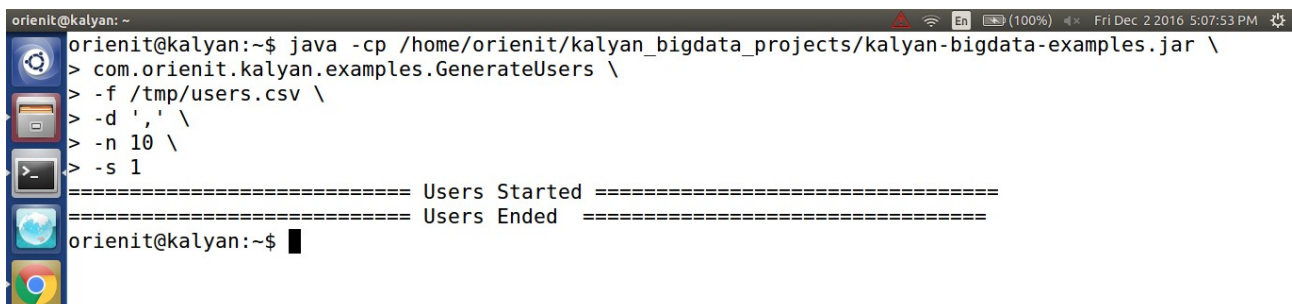
```
orienit@kalyan: ~  
orienit@kalyan:~$ mkdir /home/orienit/kalyan_bigdata_projects  
orienit@kalyan:~$
```

ii) Copy '**kalyan-bigdata-examples.jar**' jar file into '**/home/orienit/kalyan_bigdata_projects**' folder



iii) Execute Below Command to Generate Sample CSV data with 100 lines. Increase this number to get more data ...

```
java -cp /home/orienit/kalyan_bigdata_projects/kalyan-bigdata-examples.jar \  
com.orienit.kalyan.examples.GenerateUsers \  
-f /tmp/users.csv \  
-d ',' \  
-n 100 \  
-s 1
```



```
orienit@kalyan: ~  
orienit@kalyan:~$ java -cp /home/orienit/kalyan_bigdata_projects/kalyan-bigdata-examples.jar \  
> com.orienit.kalyan.examples.GenerateUsers \  
> -f /tmp/users.csv \  
> -d ',' \  
> -n 10 \  
> -s 1  
===== Users Started =====  
===== Users Ended =====  
orienit@kalyan:~$
```

6. Verify the Sample CSV data in Console, using below command

`cat /tmp/users.csv`

```
orienit@kalyan: ~$ cat /tmp/users.csv
1,user1,user1,user1@gmail.com,US,Washington,Seattle,2016-07-02 05:07:48
2,user2,user2,user2@gmail.com,US,Florida,Orlando,2016-07-02 05:07:48
3,user3,user3,user3@gmail.com,US,New York,Little Falls,2016-07-02 05:07:49
4,user4,user4,user4@gmail.com,India,Karnataka,Mangaluru,2016-07-02 05:07:49
5,user5,user5,user5@gmail.com,US,Hawaii,Hanapepe,2016-07-02 05:07:49
6,user6,user6,user6@gmail.com,India,Chennai,Kottur,2016-07-02 05:07:49
7,user7,user7,user7@gmail.com,India,Andhra Pradesh,Kakinada,2016-07-02 05:07:49
8,user8,user8,user8@gmail.com,US,Hawaii,East Honolulu,2016-07-02 05:07:49
9,user9,user9,user9@gmail.com,US,Florida,Hollywood,2016-07-02 05:07:49
10,user10,user10,user10@gmail.com,US,Washington,Bellevue,2016-07-02 05:07:49
orienit@kalyan: ~$
```

7. To work with **Flume + Hbase Integration**, Follow the below steps

i) Start the hbase using below '**start-hbase.sh**' command.

```
orienit@kalyan: ~$ start-hbase.sh
localhost: starting zookeeper, logging to /home/orienit/work/hbase-0.98.4-hadoop2/bin/../logs/hbase-
orienit-zookeeper-kalyan.out
starting master, logging to /home/orienit/work/hbase-0.98.4-hadoop2/logs/hbase-orienit-master-kalya
n.out
localhost: starting regionserver, logging to /home/orienit/work/hbase-0.98.4-hadoop2/bin/../logs/hb
ase-orienit-regionserver-kalyan.out
orienit@kalyan: ~$
```

ii. verify the hbase is running or not with "**jps**" command

```
orienit@kalyan: ~$ jps
13904 DataNode
24529 HQuorumPeer
24835 HRegionServer
14259 ResourceManager
24596 HMaster
13749 NameNode
20725 Application
14392 NodeManager
14104 SecondaryNameNode
25486 Jps
7183 org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
orienit@kalyan: ~$
```

iii. connect to hbase using '**hbase shell**' command

```
orienit@kalyan: ~$ hbase shell
2016-10-06 04:56:49,251 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. In
stead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.4-hadoop2, r890e852ce1c51b71ad180f626b71a2a1009246da, Mon Jul 14 19:45:06 PDT 2014
hbase(main):001:0>
```

iv. list out all the tables in hbase using 'list' command

```
orientit@kalyan: ~
hbase(main):002:0> list
TABLE
0 row(s) in 0.0230 seconds
=> []
hbase(main):003:0> █
```

v. create the hbase table name is 'users3' with column family name is 'cf' using below command.

create 'users3', 'cf'

```
orientit@kalyan: ~
hbase(main):002:0> create 'users3', 'cf'
0 row(s) in 2.5010 seconds
=> Hbase::Table - users3
hbase(main):003:0> █
```

vi. read the data from hbase table 'users3' using below command.

scan 'users3'

```
orientit@kalyan: ~
hbase(main):004:0> scan 'users3'
ROW COLUMN+CELL
0 row(s) in 0.0160 seconds
hbase(main):005:0> █
```

8. Execute the below command to `Extract data from CSV data into HBase using Flume`

```
$FLUME_HOME/bin/flume-ng agent -n agent --conf $FLUME_HOME/conf -f
$FLUME_HOME/conf/kalyan-csv-hbase-agent.conf -Dflume.root.logger=DEBUG,console
```

```
orientit@kalyan: ~
orientit@kalyan:~$ $FLUME_HOME/bin/flume-ng agent -n agent --conf $FLUME_HOME/conf -f $FLUME_HOME/co
nf/kalyan-csv-hbase-agent.conf -Dflume.root.logger=DEBUG,console
Info: Sourcing environment configuration script /home/orientit/work/apache-flume-1.6.0-bin/conf/flum
e-env.sh
Info: Including Hadoop libraries found via (/home/orientit/work/hadoop-2.6.0/bin/hadoop) for HDFS ac
cess
```

9. Verify the data in console


```
orientit@kalyan: ~
2016-12-27 21:23:08,485 (lifecycleSupervisor-1-3-SendThread(localhost:2181)) [INFO - org.apache.zoo
keeper.ClientCnxn$SendThread.onConnected(ClientCnxn.java:1235)] Session establishment complete on s
erver localhost/127.0.0.1:2181, sessionId = 0x15940fb2250000a, negotiated timeout = 90000
2016-12-27 21:23:08,492 (lifecycleSupervisor-1-3-SendThread(localhost:2181)) [DEBUG - org.apache.zo
ookeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionId:0x15940fb2
250000a, packet:: clientPath:null serverPath:null finished:false header:: 1,3 replyHeader:: 1,75,0
request:: '/hbase/hbaseid,F response:: s{13,13,1482853925454,1482853925454,0,0,0,0,67,0,13}
2016-12-27 21:23:08,495 (lifecycleSupervisor-1-3-SendThread(localhost:2181)) [DEBUG - org.apache.zo
ookeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionId:0x15940fb2
250000a, packet:: clientPath:null serverPath:null finished:false header:: 2,4 replyHeader:: 2,75,0
request:: '/hbase/hbaseid,F response:: #ffffffff000146d61737465723a3630303030fffffe2515c14d5930
7650425546a2430386637343961632d626562642d343661342d386635662d663134373061363361623563,s{13,13,14828
53925454,1482853925454,0,0,0,0,67,0,13}
2016-12-27 21:23:09,244 (lifecycleSupervisor-1-3) [INFO - org.apache.hadoop.conf.Configuration.warn
OnceIfDeprecated(Configuration.java:1049)] hadoop.native.lib is deprecated. Instead, use io.native.
lib.available
2016-12-27 21:23:09,412 (lifecycleSupervisor-1-3-SendThread(localhost:2181)) [DEBUG - org.apache.zo
ookeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionId:0x15940fb2
250000a, packet:: clientPath:null serverPath:null finished:false header:: 3,3 replyHeader:: 3,75,0
request:: '/hbase,F response:: s{3,3,1482853923101,1482853923101,0,15,0,0,0,15,51}
2016-12-27 21:23:09,419 (lifecycleSupervisor-1-3-SendThread(localhost:2181)) [DEBUG - org.apache.zo
ookeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionId:0x15940fb2
250000a, packet:: clientPath:null serverPath:null finished:false header:: 4,4 replyHeader:: 4,75,0
request:: '/hbase/master,F response:: #ffffffff000146d61737465723a363030303017fffff319663fffff
ff89fffffd43350425546a16a96c6f63616c686f737410fffffe0fffffd4318fffffcdfffffecfffff87f
ffff942b100,s{12,12,1482853923967,1482853923967,0,0,0,0,97180314321813505,55,0,12}
2016-12-27 21:23:09,641 (lifecycleSupervisor-1-3) [INFO - org.apache.flume.instrumentation.Monitore
```

10. Verify the data in HBase

Execute below command to get the data from hbase table 'users3'

count 'users3'

scan 'users3'

```
orientit@kalyan: ~
hbase(main):004:0> count 'users3'
10 row(s) in 0.0220 seconds

=> 10
hbase(main):005:0> scan 'users3'
ROW COLUMN+CELL
1482853991034-d6E0iSyaFm column=cf:city, timestamp=1482853994225, value=Canandaigua
-0-
1482853991034-d6E0iSyaFm column=cf:country, timestamp=1482853994225, value=US
-0-
1482853991034-d6E0iSyaFm column=cf:dt, timestamp=1482853994225, value=2016-43-27 08:43:39
-0-
1482853991034-d6E0iSyaFm column=cf:email, timestamp=1482853994225, value=user91@gmail.com
-0-
1482853991034-d6E0iSyaFm column=cf:password, timestamp=1482853994225, value=user91
-0-
1482853991034-d6E0iSyaFm column=cf:state, timestamp=1482853994225, value=New York
-0-
1482853991034-d6E0iSyaFm column=cf:useridd, timestamp=1482853994225, value=91
-0-
1482853991034-d6E0iSyaFm column=cf:username, timestamp=1482853994225, value=user91
-0-
1482853991048-d6E0iSyaFm column=cf:city, timestamp=1482853994225, value=Glens Falls
-1-
1482853991048-d6E0iSyaFm column=cf:country, timestamp=1482853994225, value=US
-1-
1482853991048-d6E0iSyaFm column=cf:dt, timestamp=1482853994225, value=2016-43-27 08:43:39
```