

## **Kalyan Big Data Projects – Project 13 How To Stream REGEX 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>)

`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-regex-hbase-agent.conf` ==> [link](https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/flume/project13-hbase-regex/kalyan-regex-hbase-agent.conf)  
(<https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/flume/project13-hbase-regex/kalyan-regex-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. Complex Data is the output format then REGEX is best solution
  - We can use Hbase to analyze this data
- 

1. create "**kalyan-regex-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 = users1
agent.sinks.HBASE.columnFamily = cf
agent.sinks.HBASE.serializer = org.apache.flume.sink.hbase.RegexHbaseEventSerializer
agent.sinks.HBASE.serializer.regex = ^([^\,]*),([^\,]*),([^\,]*),([^\,]*),([^\,]*),([^\,]*),([^\,]*),([^\,]*)$
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-regex-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

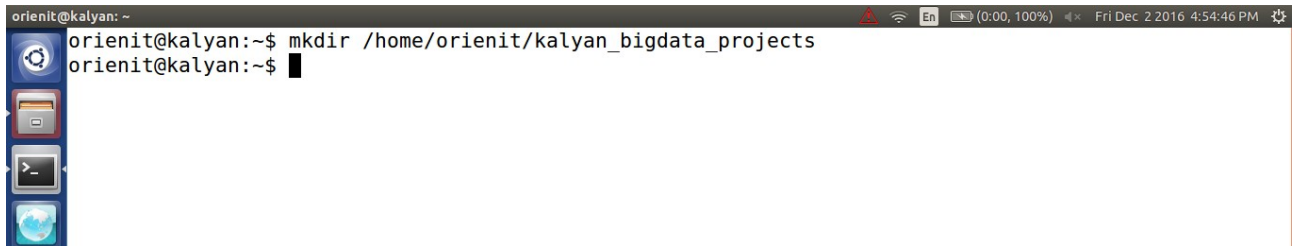
4. Generate Large Amount of Sample CSV data follow this [article](http://kalyanbigdatatraining.blogspot.com/2016/12/how-to-generate-large-amount-of-sample.html).

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

5. Follow below steps...

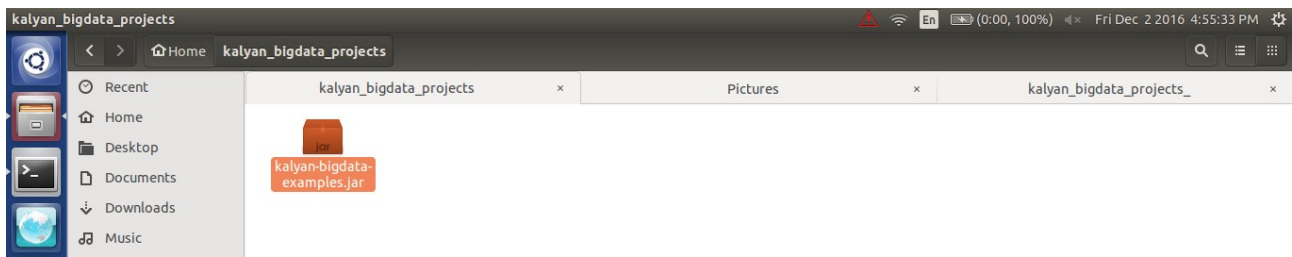
i) Create '**kalyan\_bigdata\_projects**' folder in **user home** (i.e **/home/orienit**)

**Command:** `mkdir /home/orienit/kalyan_bigdata_projects`



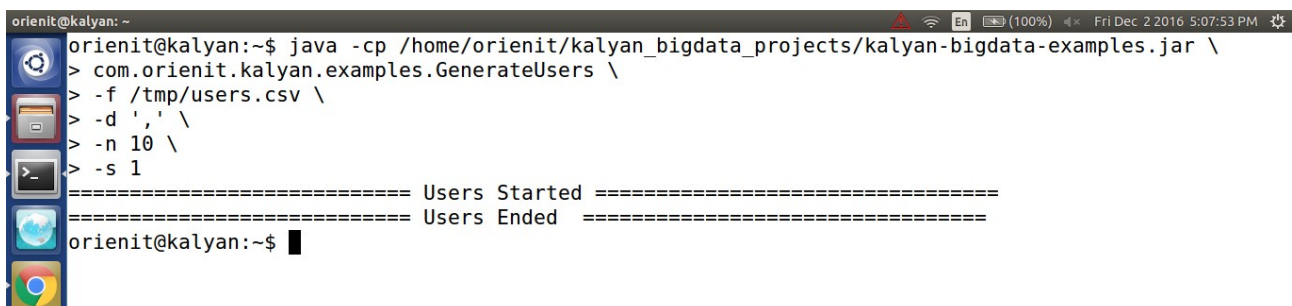
```
orienit@kalyan: ~  
orienit@kalyan:~$ mkdir /home/orienit/kalyan_bigdata_projects  
orienit@kalyan:~$
```

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: ~  
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: ~  
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: ~  
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

```
orientit@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

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

5. create the hbase table name is 'users1' with column family name is 'cf' using below command.

create 'users1', 'cf'

```
hbase(main):003:0> create 'users1' , 'cf'
0 row(s) in 0.8880 seconds
=> Hbase::Table - users1
hbase(main):004:0>
```

6. read the data from hbase table 'users1' using below scan 'users1' command.

```
hbase(main):004:0> scan 'users1'
ROW COLUMN+CELL
0 row(s) in 0.0510 seconds
hbase(main):005:0>
```

7. 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-regex-hbase-agent.conf -Dflume.root.logger=DEBUG,console

```
orientit@kalyan: ~$ FLUME_HOME/bin/flume-ng agent -n agent --conf $FLUME_HOME/conf -f $FLUME_HOME/co
nf/kalyan-regex-hbase-agent.conf -Dflume.root.logger=DEBUG,console
Info: Including Hadoop libraries found via (/home/orientit/work/hadoop-2.6.0/bin/hadoop) for HDFS ac
cess
Info: Excluding /home/orientit/work/hadoop-2.6.0/share/hadoop/common/lib/slf4j-api-1.7.5.jar from cl
asspath
Info: Excluding /home/orientit/work/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar fro
m classpath
```



**8. Verify the data in console**

```
orientit@kalyan: ~  
fffb9fffff92a100,s{256,256,1475709972484,1475709972484,0,0,0,96712128430800896,55,0,256}  
2016-10-06 05:10:24,725 (LifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.register(MonitoredCounterGroup.java:120)] Monitored counter group for type: SINK, name: HBASE: Successfully registered new MBean.  
2016-10-06 05:10:24,725 (LifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.start(MonitoredCounterGroup.java:96)] Component type: SINK, name: HBASE started  
2016-10-06 05:10:24,726 (SinkRunner-PollingRunner-DefaultSinkProcessor) [DEBUG - org.apache.flume.SinkRunner$PollingRunner.run(SinkRunner.java:143)] Polling sink runner starting  
2016-10-06 05:10:29,244 (LifecycleSupervisor-1-1-SendThread(localhost:2181)) [DEBUG - org.apache.zookeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionId:0x157972b1c07000b, packet: clientPath:null, serverPath:null, finished:false, headers: 5, 4, replyHeader: 5, 202
```

**9. Verify the data in HBase**

Execute below command to get the data from hbase table '**users1**'

count 'users1'

scan 'users1'

```
orientit@kalyan: ~  
hbase(main):004:0> scan 'users1'  
ROW COLUMN+CELL  
0 row(s) in 0.0510 seconds  
  
hbase(main):005:0> count 'users1'  
10 row(s) in 0.0430 seconds  
  
=> 10  
hbase(main):006:0> scan 'users1'  
ROW COLUMN+CELL  
1475710826216-x14v0BAnAc column=cf:city, timestamp=1475710829391, value=Hornell  
-0  
1475710826216-x14v0BAnAc column=cf:country, timestamp=1475710829391, value=US  
-0  
1475710826216-x14v0BAnAc column=cf:date, timestamp=1475710829391, value=2016-06-06 05:06:15  
-0  
1475710826216-x14v0BAnAc column=cf:email, timestamp=1475710829391, value=user91@gmail.com  
-0  
1475710826216-x14v0BAnAc column=cf:password, timestamp=1475710829391, value=user91  
-0  
1475710826216-x14v0BAnAc column=cf:state, timestamp=1475710829391, value=New York  
-0  
1475710826216-x14v0BAnAc column=cf:useridd, timestamp=1475710829391, value=91  
-0  
1475710826216-x14v0BAnAc column=cf:username, timestamp=1475710829391, value=user91  
-0
```