Kalyan Big Data Projects – Project 12
How To Stream JSON 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)

`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)

`hbase-1.1.2-bin.tar.gz` ==> link (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)

`kalyan-bigdata-examples.jar` ==> link (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)

`kalyan-json-hbase-agent.conf` ==> link (https://github.com/kalyanhadooptraining/kalyan-bigdata-realtime-projects/blob/master/flume/project12-hbase-json/kalyan-json-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. JSON is the output format
- We can use Hbase to analyze this data

---

1. create "**kalyan-json-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.json
agent.sources.EXEC.channels = MemChannel

agent.sinks.HBASE.type = hbase
agent.sinks.HBASE.table = users2
agent.sinks.HBASE.columnFamily = cf
agent.sinks.HBASE.serializer = com.orienit.kalyan.flume.sink.JsonHbaseEventSerializer
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-json-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

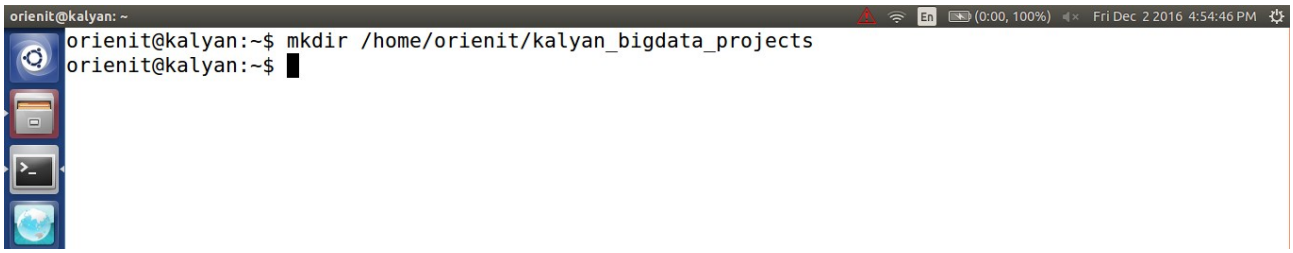
4. Generate Large Amount of Sample JSON data follow this article.

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

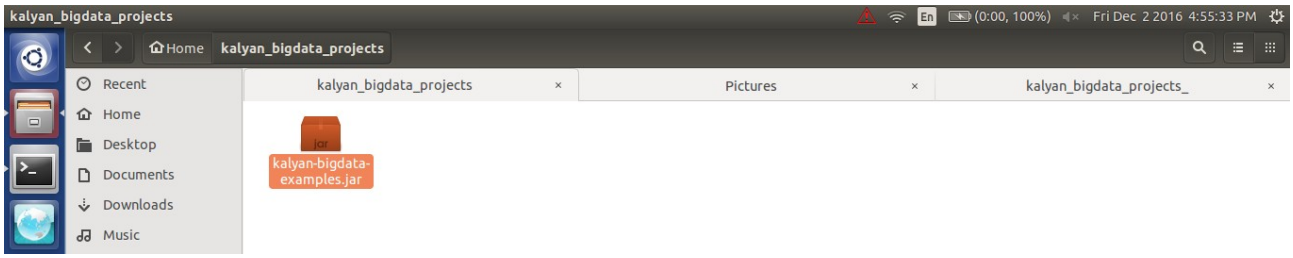
5. Follow below steps...

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

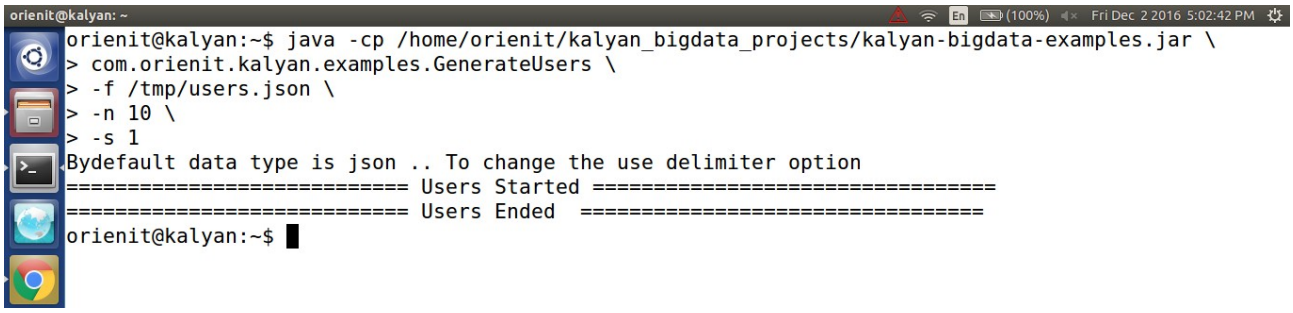
Command: *mkdir /home/orienit/kalyan_bigdata_projects*



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



iii) Execute below command to Generate Sample JSON 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.json \
-n 100 \
-s 1
```

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

cat /tmp/users.json

```
orienit@kalyan:~$ cat /tmp/users.json
{"userid":1,"username":"user1","password":"user1","email":"user1@gmail.com","country":"India","stat
e":"Andhra Pradesh","city":"Guntur","dt":"2016-02-02 05:02:39"}
{"userid":2,"username":"user2","password":"user2","email":"user2@gmail.com","country":"US","state":
"Washington","city":"Renton","dt":"2016-02-02 05:02:39"}
{"userid":3,"username":"user3","password":"user3","email":"user3@gmail.com","country":"US","state":
"Hawaii","city":"Hanapepe","dt":"2016-02-02 05:02:39"}
{"userid":4,"username":"user4","password":"user4","email":"user4@gmail.com","country":"US","state":
"Washington","city":"Bellingham","dt":"2016-02-02 05:02:39"}
{"userid":5,"username":"user5","password":"user5","email":"user5@gmail.com","country":"India","stat
e":"Andhra Pradesh","city":"Kakinada","dt":"2016-02-02 05:02:39"}
{"userid":6,"username":"user6","password":"user6","email":"user6@gmail.com","country":"India","stat
e":"Telangana","city":"Karimnagar","dt":"2016-02-02 05:02:39"}
{"userid":7,"username":"user7","password":"user7","email":"user7@gmail.com","country":"US","state":
"New York","city":"Albany","dt":"2016-02-02 05:02:40"}
{"userid":8,"username":"user8","password":"user8","email":"user8@gmail.com","country":"India","stat
e":"Karnataka","city":"Bidar","dt":"2016-02-02 05:02:40"}
{"userid":9,"username":"user9","password":"user9","email":"user9@gmail.com","country":"India","stat
e":"Chennai","city":"Virugambakkam","dt":"2016-02-02 05:02:40"}
{"userid":10,"username":"user10","password":"user10","email":"user10@gmail.com","country":"US","sta
te":"Hawaii","city":"Honolulu","dt":"2016-02-02 05:02:40"}
```

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

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

=> []
hbase(main):003:0>
```

v. create the hbase table name is '**users2**' with column family name is '**cf**' using below command.

create 'users2', 'cf'

```
hbase(main):011:0> create 'users2', 'cf'
0 row(s) in 0.7440 seconds

=> Hbase::Table - users2
hbase(main):012:0>
```

vi. read the data from hbase table '**users2**' using below command.

scan 'users2'

```
hbase(main):012:0> scan 'users2'
ROW                                    COLUMN+CELL
0 row(s) in 0.0200 seconds

hbase(main):013:0>
```

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

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

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

9. Verify the data in console

```
orienit@kalyan: ~
ffb9ffffff92a100,s{256,256,1475709972484,1475709972484,0,0,0,96712128430800896,55,0,256}
2016-10-06 05:43:43,626 (lifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.Monitore
dCounterGroup.register(MonitoredCounterGroup.java:120)] Monitored counter group for type: SINK, nam
e: HBASE: Successfully registered new MBean.
2016-10-06 05:43:43,626 (lifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.Monitore
dCounterGroup.start(MonitoredCounterGroup.java:96)] Component type: SINK, name: HBASE started
2016-10-06 05:43:43,627 (SinkRunner-PollingRunner-DefaultSinkProcessor) [DEBUG - org.apache.flume.S
inkRunner$PollingRunner.run(SinkRunner.java:143)] Polling sink runner starting
2016-10-06 05:43:48,114 (lifecycleSupervisor-1-1-SendThread(localhost:2181)) [DEBUG - org.apache.zo
okeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionid:0x157972b1
c970016, packet:: clientPath:null serverPath:null finished:false header:: 5,4  replyHeader:: 5,349,
0  request:: '/hbase/meta-region-server,F  response:: #ffffffff0001a726567696f6e7365727665723a36303
03230106174c4a6ffffff8ffffff8650425546a16a96c6f63616c686f737410ffffff4ffffffd4318ffffff9effffffd
4ffffffacffffffb9ffffff92a100,s{272,272,1475709978687,1475709978687,0,0,0,0,61,0,272}
2016-10-06 05:43:48,132 (lifecycleSupervisor-1-1-SendThread(localhost:2181)) [DEBUG - org.apache.zo
okeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionid:0x157972b1
c970016, packet:: clientPath:null serverPath:null finished:false header:: 6,4  replyHeader:: 6,349,
0  request:: '/hbase/meta-region-server,F  response:: #ffffffff0001a726567696f6e7365727665723a36303
03230106174c4a6ffffff8ffffff8650425546a16a96c6f63616c686f737410ffffff4ffffffd4318ffffff9effffffd
4ffffffacffffffb9ffffff92a100,s{272,272,1475709978687,1475709978687,0,0,0,0,61,0,272}
2016-10-06 05:43:48,210 (lifecycleSupervisor-1-1-SendThread(localhost:2181)) [DEBUG - org.apache.zo
okeeper.ClientCnxn$SendThread.readResponse(ClientCnxn.java:818)] Reading reply sessionid:0x157972b1
c970016, packet:: clientPath:null serverPath:null finished:false header:: 7,4  replyHeader:: 7,349,
0  request:: '/hbase/meta-region-server,F  response:: #ffffffff0001a726567696f6e7365727665723a36303
```

10. Verify the data in HBase

Execute below command to get the data from hbase table '**users2**'

count 'users2'

scan 'users2'

```
orienit@kalyan: ~

hbase(main):012:0> scan 'users2'
ROW                       COLUMN+CELL
0 row(s) in 0.0200 seconds

hbase(main):013:0> count 'users2'
10 row(s) in 0.0150 seconds

=> 10
hbase(main):014:0> scan 'users2'
ROW                       COLUMN+CELL
 1475712825060-p4b9Uai7tz column=cf:city, timestamp=1475712828286, value=Tirupati
 -0
 1475712825060-p4b9Uai7tz column=cf:country, timestamp=1475712828286, value=India
 -0
 1475712825060-p4b9Uai7tz column=cf:date, timestamp=1475712828286, value=2016-37-06 05:37:54
 -0
 1475712825060-p4b9Uai7tz column=cf:email, timestamp=1475712828286, value=user91@gmail.com
 -0
 1475712825060-p4b9Uai7tz column=cf:password, timestamp=1475712828286, value=user91
 -0
 1475712825060-p4b9Uai7tz column=cf:state, timestamp=1475712828286, value=Andhra Pradesh
 -0
 1475712825060-p4b9Uai7tz column=cf:userid, timestamp=1475712828286, value=91
 -0
 1475712825060-p4b9Uai7tz column=cf:username, timestamp=1475712828286, value=user91
 -0
```