CS-523DE-Big-Data-Technology-Project

1. Iyosiyas Workie Mitiku - 614166
2. El Mehdi Korda - 614134

Agenda

1.Project Description

CSV Data

- player_country
- first_name
- String last_name
- goals_scored
- champions_league_matches_played
- english_league_matches_played
- minutes_played
- assists
- tackles
- age

Technologies

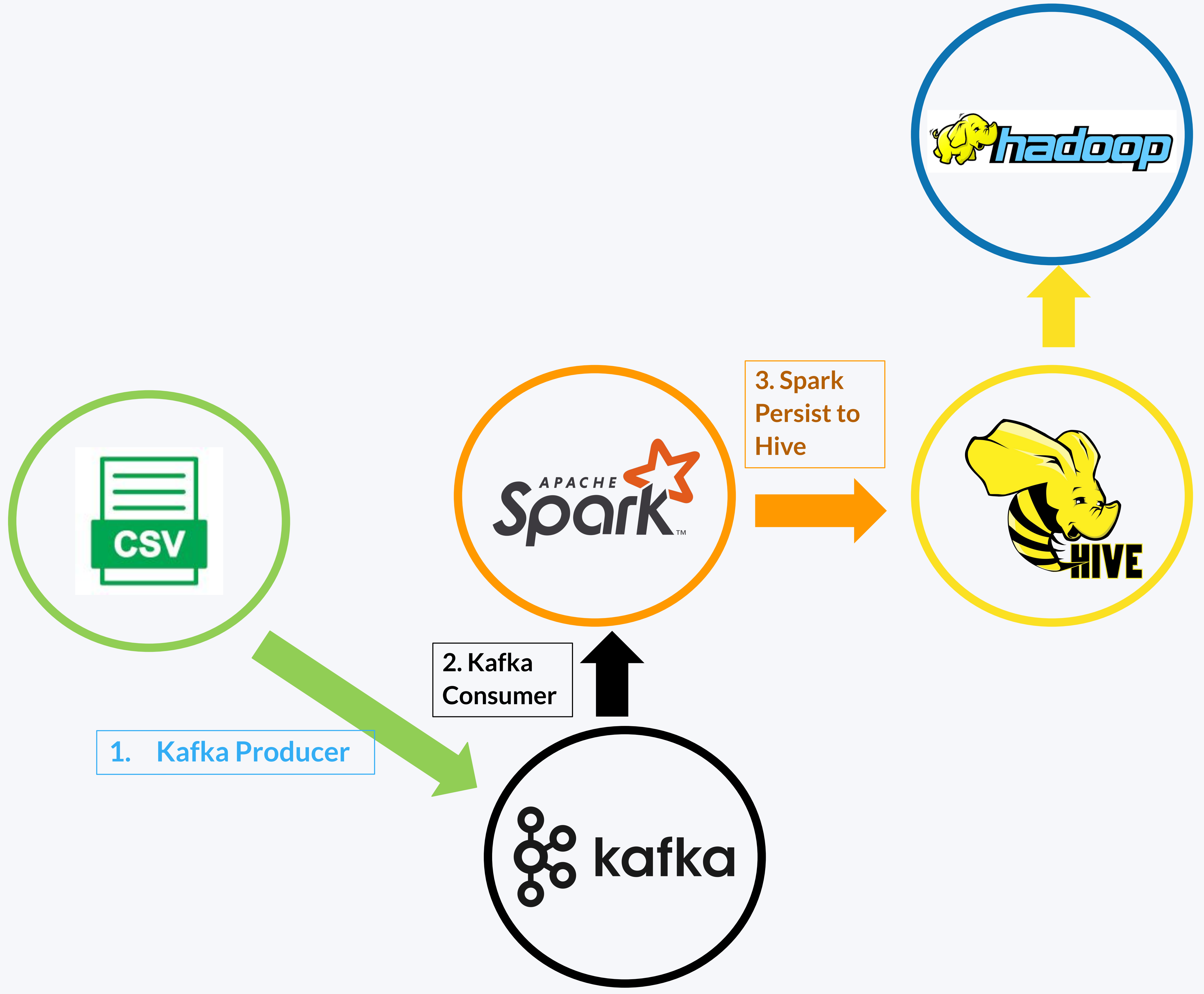
Apache Spark

kafka

HIVE

hadoop

1. Spark Streaming - Data Flow

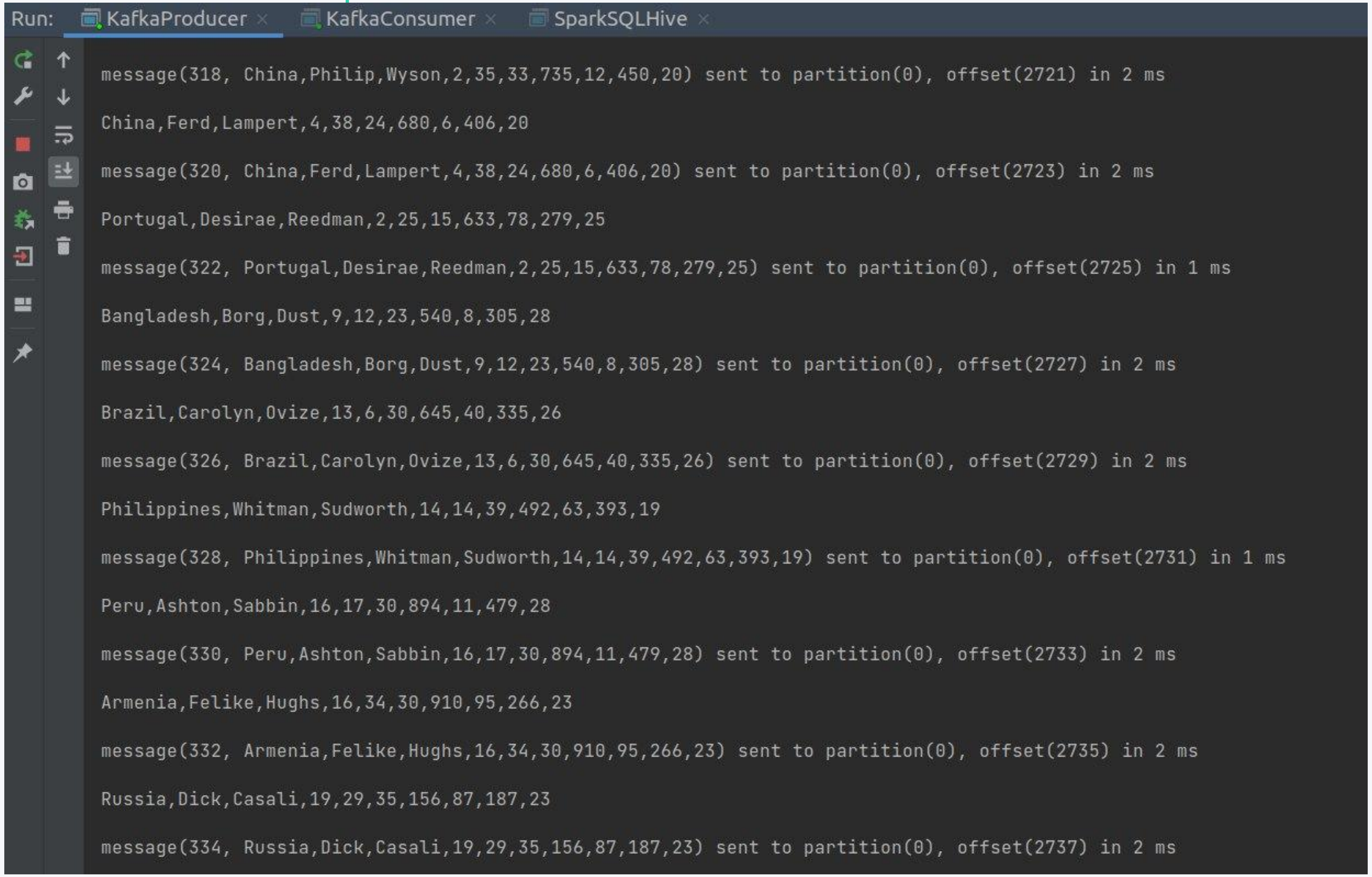


1.1 Kafka Producer

kafka

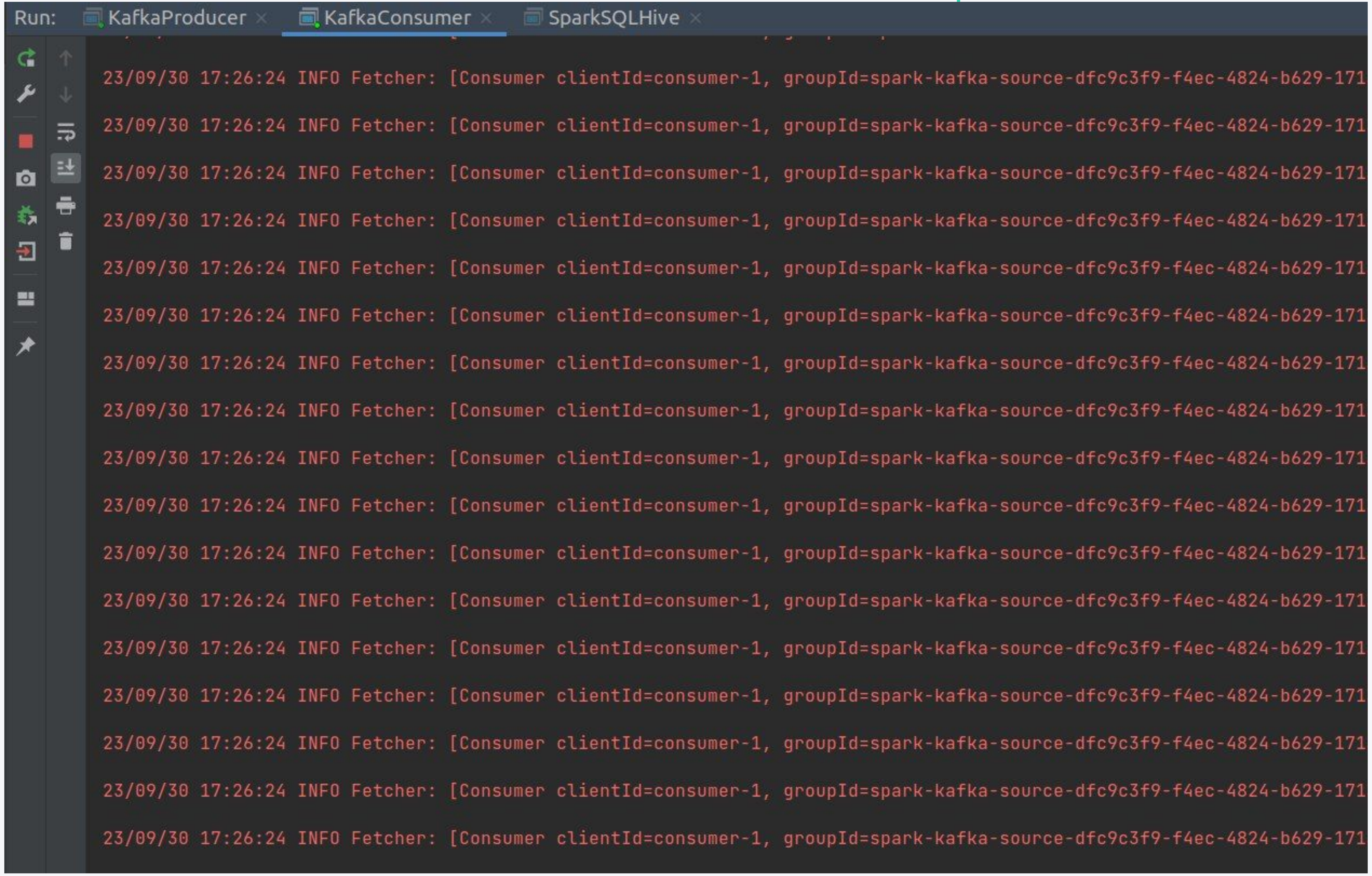
- We have used Kafka as a broker to produce messages that we get from our data source

- As seen in the screenshot the Footballer data we get is being produced to a kafka topic we specified



```
message(318, China,Philip,Wyson,2,35,33,735,12,450,20) sent to partition(0), offset(2721) in 2 ms
China,Ferd,Lampert,4,38,24,680,6,406,20
message(320, China,Ferd,Lampert,4,38,24,680,6,406,20) sent to partition(0), offset(2723) in 2 ms
Portugal,Desirae,Reedman,2,25,15,633,78,279,25
message(322, Portugal,Desirae,Reedman,2,25,15,633,78,279,25) sent to partition(0), offset(2725) in 1 ms
Bangladesh,Borg,Dust,9,12,23,540,8,305,28
message(324, Bangladesh,Borg,Dust,9,12,23,540,8,305,28) sent to partition(0), offset(2727) in 2 ms
Brazil,Carolyn,Ovize,13,6,30,645,40,335,26
message(326, Brazil,Carolyn,Ovize,13,6,30,645,40,335,26) sent to partition(0), offset(2729) in 2 ms
Philippines,Whitman,Sudworth,14,14,39,492,63,393,19
message(328, Philippines,Whitman,Sudworth,14,14,39,492,63,393,19) sent to partition(0), offset(2731) in 1 ms
Peru,Ashton,Sabbin,16,17,30,894,11,479,28
message(330, Peru,Ashton,Sabbin,16,17,30,894,11,479,28) sent to partition(0), offset(2733) in 2 ms
Armenia,Felike,Hughs,16,34,30,910,95,266,23
message(332, Armenia,Felike,Hughs,16,34,30,910,95,266,23) sent to partition(0), offset(2735) in 2 ms
Russia,Dick,Casali,19,29,35,156,87,187,23
message(334, Russia,Dick,Casali,19,29,35,156,87,187,23) sent to partition(0), offset(2737) in 2 ms
```

1.1 Kafka Consumer



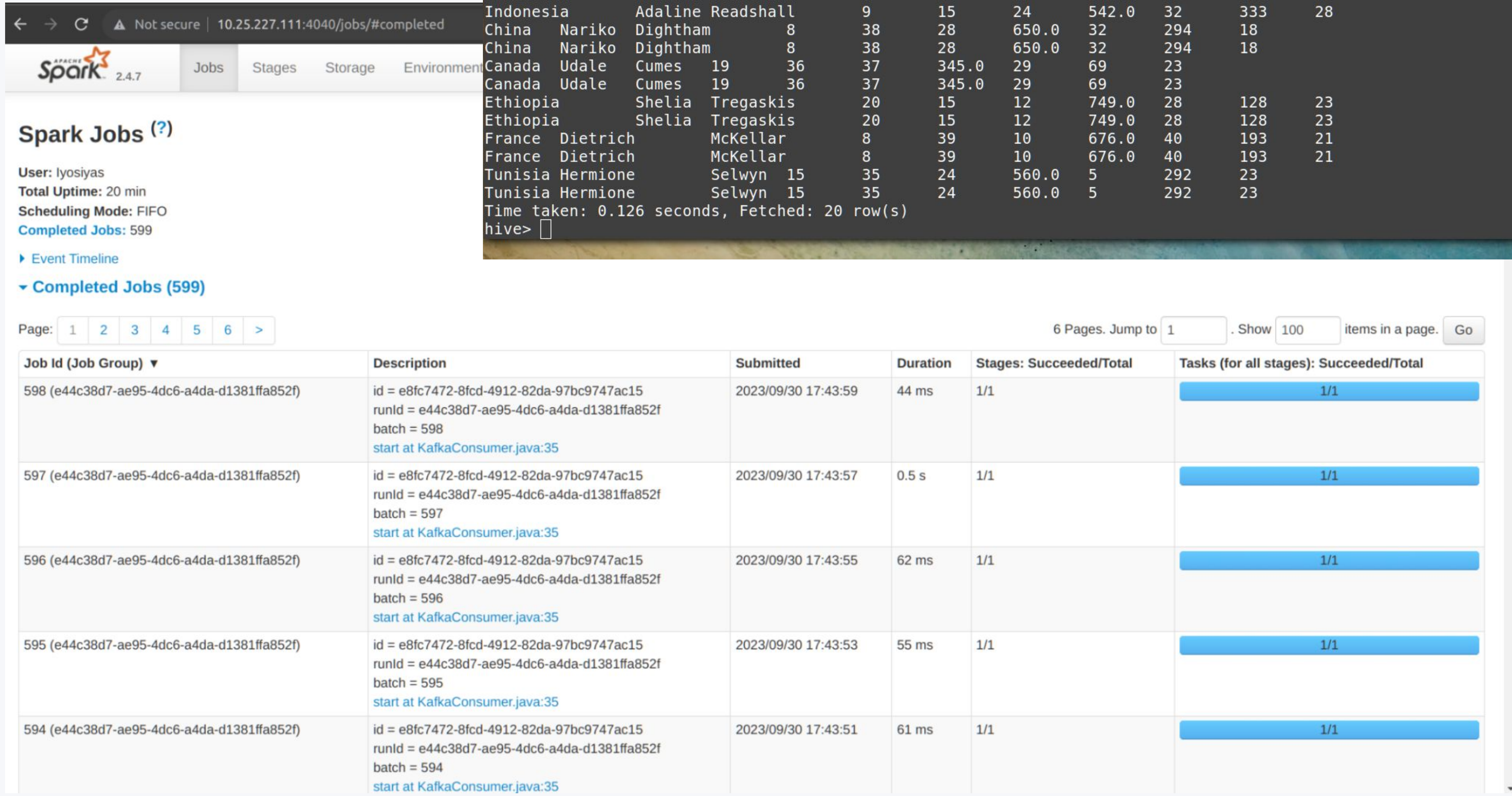
```
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
23/09/30 17:26:24 INFO Fetcher: [Consumer clientId=consumer-1, groupId=spark-kafka-source-dfc9c3f9-f4ec-4824-b629-171
```

- As seen in the image we are listening to what we produced

- After listening from a topic we are write it to spark

2. Spark Streaming - Integration with Hive

- As seen in the screenshot we are using Spark to process data stored in Hive.
- This integration allows users to take advantage of Spark's performance and scalability for Hive workloads.



Spark Jobs (?)

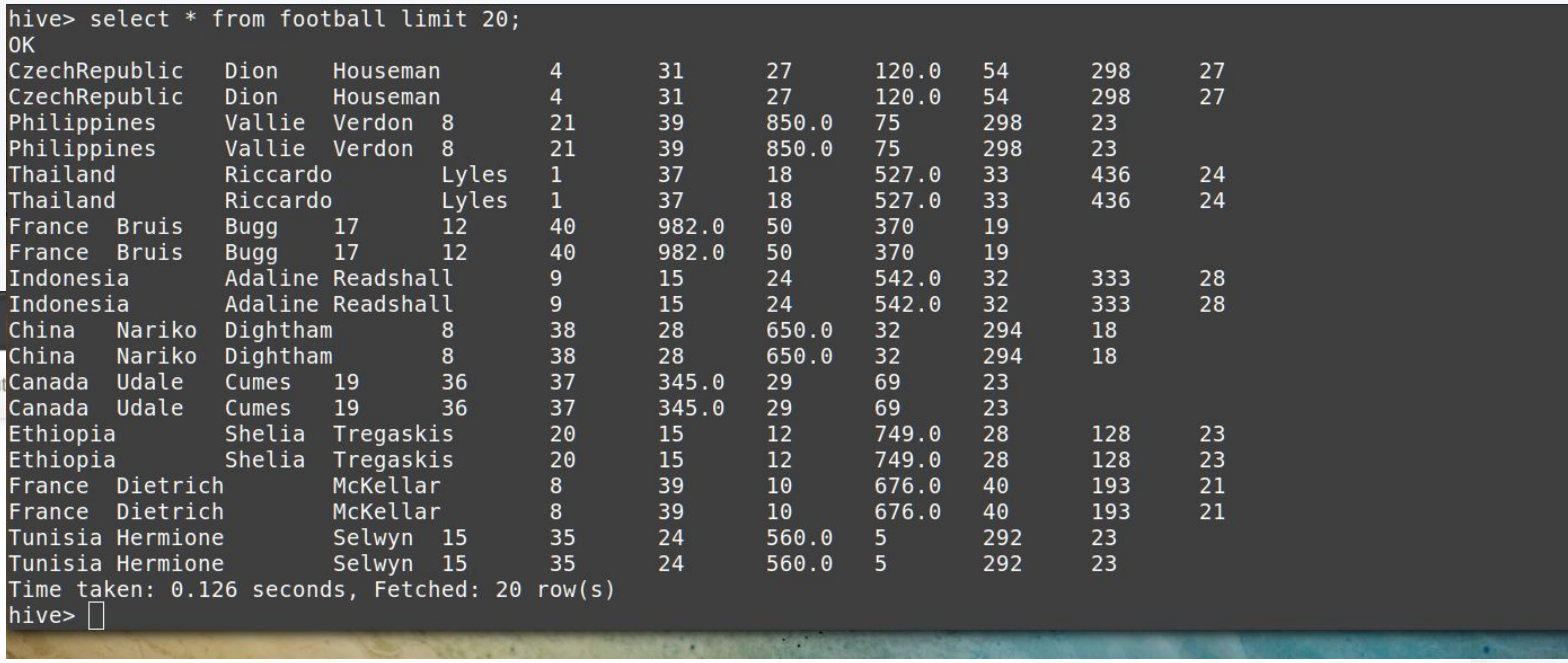
User: lyosiyas
Total Uptime: 20 min
Scheduling Mode: FIFO
Completed Jobs: 599

▸ Event Timeline

▾ Completed Jobs (599)

Page: 1 2 3 4 5 6 >   6 Pages. Jump to 1 . Show 100 items in a page. Go

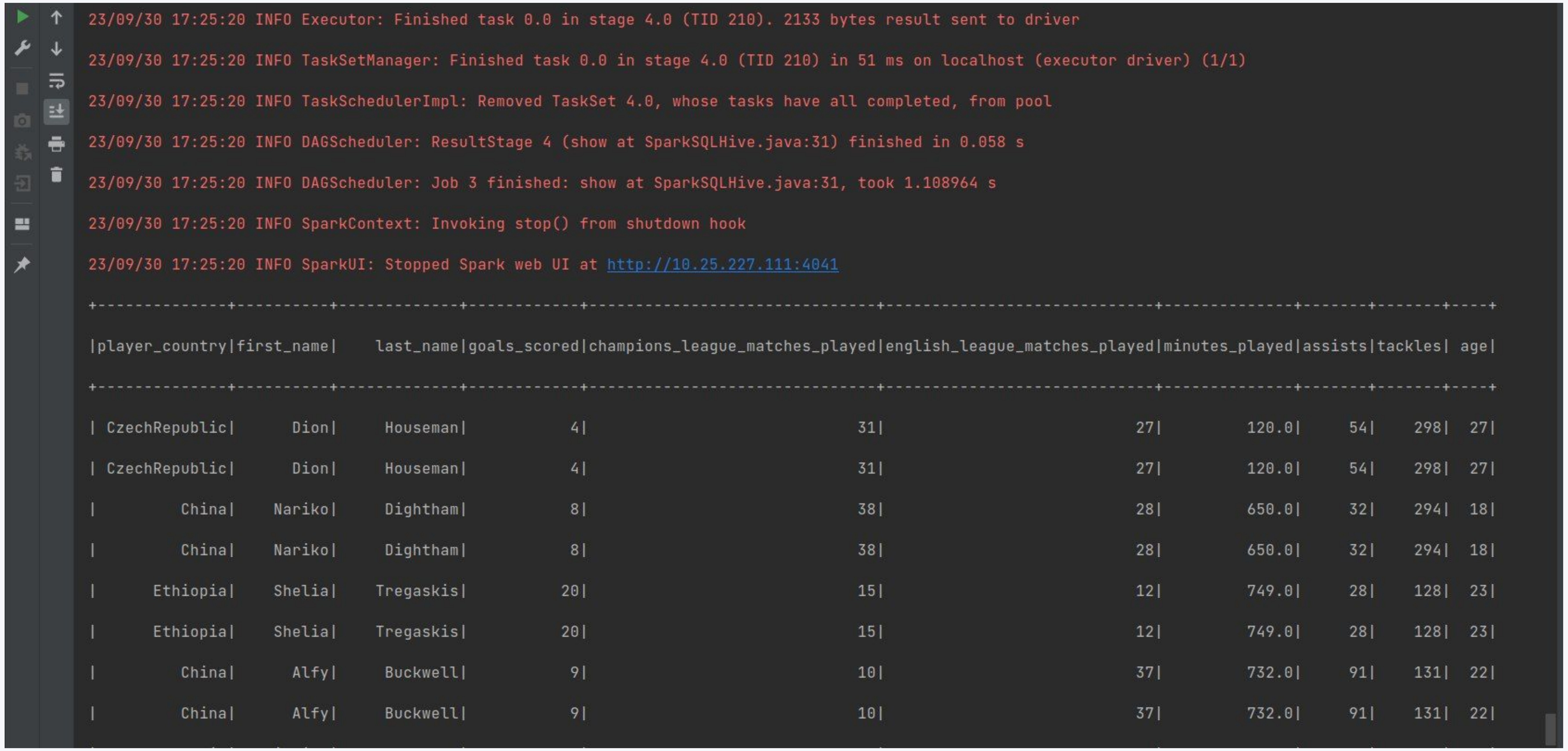
Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
598 (e44c38d7-ae95-4dc6-a4da-d1381ffa852f)	id = e8fc7472-8fcd-4912-82da-97bc9747ac15 runId = e44c38d7-ae95-4dc6-a4da-d1381ffa852f batch = 598 start at KafkaConsumer.java:35	2023/09/30 17:43:59	44 ms	1/1	1/1
597 (e44c38d7-ae95-4dc6-a4da-d1381ffa852f)	id = e8fc7472-8fcd-4912-82da-97bc9747ac15 runId = e44c38d7-ae95-4dc6-a4da-d1381ffa852f batch = 597 start at KafkaConsumer.java:35	2023/09/30 17:43:57	0.5 s	1/1	1/1
596 (e44c38d7-ae95-4dc6-a4da-d1381ffa852f)	id = e8fc7472-8fcd-4912-82da-97bc9747ac15 runId = e44c38d7-ae95-4dc6-a4da-d1381ffa852f batch = 596 start at KafkaConsumer.java:35	2023/09/30 17:43:55	62 ms	1/1	1/1
595 (e44c38d7-ae95-4dc6-a4da-d1381ffa852f)	id = e8fc7472-8fcd-4912-82da-97bc9747ac15 runId = e44c38d7-ae95-4dc6-a4da-d1381ffa852f batch = 595 start at KafkaConsumer.java:35	2023/09/30 17:43:53	55 ms	1/1	1/1
594 (e44c38d7-ae95-4dc6-a4da-d1381ffa852f)	id = e8fc7472-8fcd-4912-82da-97bc9747ac15 runId = e44c38d7-ae95-4dc6-a4da-d1381ffa852f batch = 594 start at KafkaConsumer.java:35	2023/09/30 17:43:51	61 ms	1/1	1/1



```
hive> select * from football limit 20;
OK
CzechRepublic   Dion    Houseman        4       31      27      120.0   54      298     27
CzechRepublic   Dion    Houseman        4       31      27      120.0   54      298     27
Philippines     Vallie  Verdon  8       21      39      850.0   75      298     23
Philippines     Vallie  Verdon  8       21      39      850.0   75      298     23
Thailand        Riccardo        Lyles   1       37      18      527.0   33      436     24
Thailand        Riccardo        Lyles   1       37      18      527.0   33      436     24
France  Bruis   Bugg    17      12      40      982.0   50      370     19
France  Bruis   Bugg    17      12      40      982.0   50      370     19
Indonesia       Adaline Readshall       9       15      24      542.0   32      333     28
Indonesia       Adaline Readshall       9       15      24      542.0   32      333     28
China   Nariko  Dightham        8       38      28      650.0   32      294     18
China   Nariko  Dightham        8       38      28      650.0   32      294     18
Canada  Udale   Cumes   19      36      37      345.0   29      69      23
Canada  Udale   Cumes   19      36      37      345.0   29      69      23
Ethiopia        Shelia  Tregaskis       20      15      12      749.0   28      128     23
Ethiopia        Shelia  Tregaskis       20      15      12      749.0   28      128     23
France  Dietrich        McKellar        8       39      10      676.0   40      193     21
France  Dietrich        McKellar        8       39      10      676.0   40      193     21
Tunisia Hermione        Selwyn  15      35      24      560.0   5       292     23
Tunisia Hermione        Selwyn  15      35      24      560.0   5       292     23
Time taken: 0.126 seconds, Fetched: 20 row(s)
hive> 
```

2. Spark Streaming - Integration with Hive

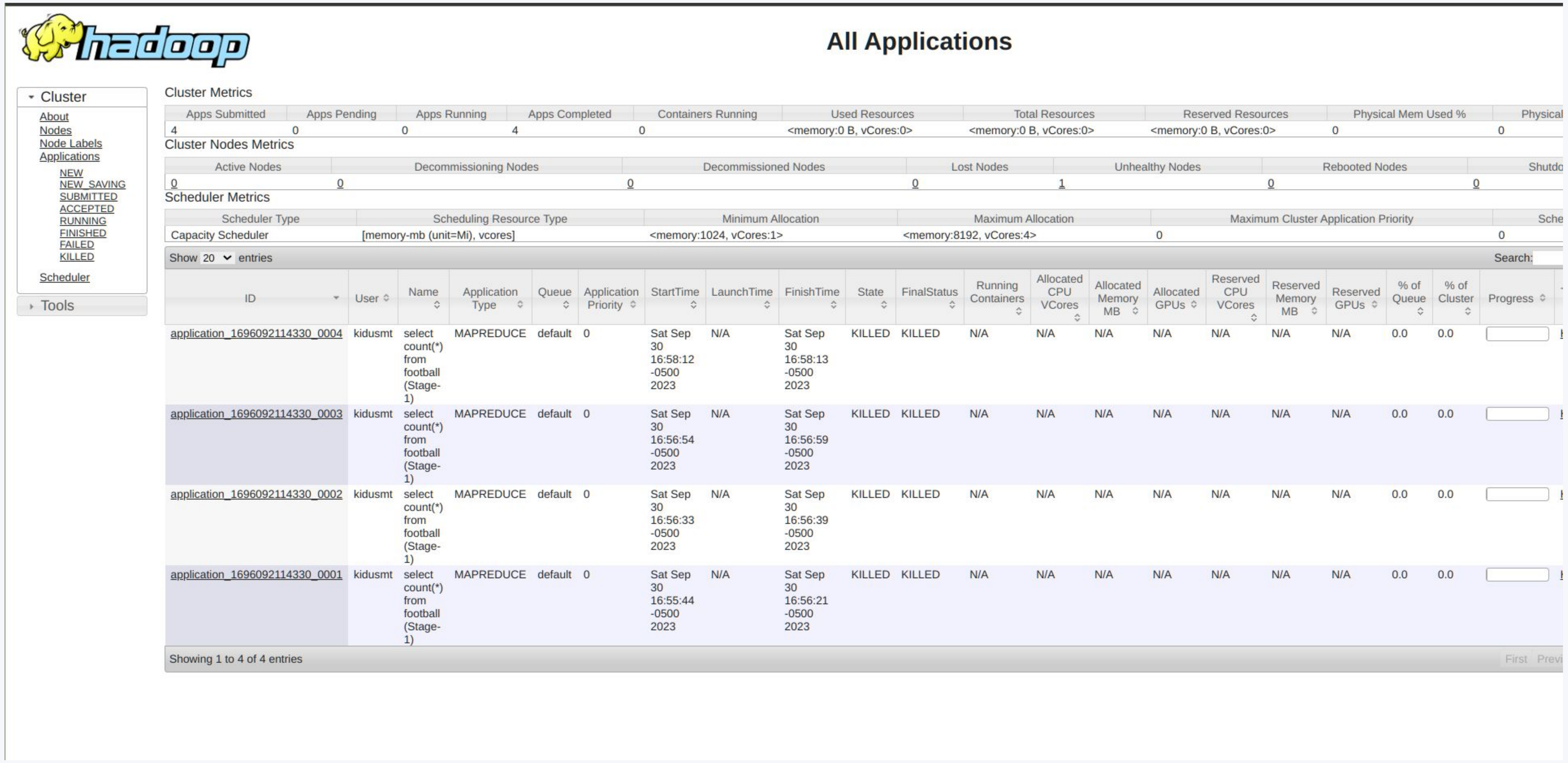
- As seen in the screenshot we are using spark to write sql messages which is then converted to mapreduce then gives the result on the right



```
23/09/30 17:25:20 INFO Executor: Finished task 0.0 in stage 4.0 (TID 210). 2133 bytes result sent to driver
23/09/30 17:25:20 INFO TaskSetManager: Finished task 0.0 in stage 4.0 (TID 210) in 51 ms on localhost (executor driver) (1/1)
23/09/30 17:25:20 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
23/09/30 17:25:20 INFO DAGScheduler: ResultStage 4 (show at SparkSQLHive.java:31) finished in 0.058 s
23/09/30 17:25:20 INFO DAGScheduler: Job 3 finished: show at SparkSQLHive.java:31, took 1.108964 s
23/09/30 17:25:20 INFO SparkContext: Invoking stop() from shutdown hook
23/09/30 17:25:20 INFO SparkUI: Stopped Spark web UI at http://10.25.227.111:4041
+-------------+----------+------------+------------+--------------------------------+------------------------------+-------------+-------+-------+----+
|player_country|first_name|    last_name|goals_scored|champions_league_matches_played|english_league_matches_played|minutes_played|assists|tackles| age|
+-------------+----------+------------+------------+--------------------------------+------------------------------+-------------+-------+-------+----+
| CzechRepublic|      Dion|    Houseman|           4|                              31|                            27|        120.0|     54|    298|  27|
| CzechRepublic|      Dion|    Houseman|           4|                              31|                            27|        120.0|     54|    298|  27|
|         China|    Nariko|    Dightham|           8|                              38|                            28|        650.0|     32|    294|  18|
|         China|    Nariko|    Dightham|           8|                              38|                            28|        650.0|     32|    294|  18|
|      Ethiopia|    Shelia|   Tregaskis|          20|                              15|                            12|        749.0|     28|    128|  23|
|      Ethiopia|    Shelia|   Tregaskis|          20|                              15|                            12|        749.0|     28|    128|  23|
|         China|      Alfy|    Buckwell|           9|                              10|                            37|        732.0|     91|    131|  22|
|         China|      Alfy|    Buckwell|           9|                              10|                            37|        732.0|     91|    131|  22|
```

3. Hadoop - Further Results Captured

- The image on the right shows our hadoop which is the underlying storage used to store our data



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources	Physical Mem Used %	Physical
4	0	0	4	0	<memory:0 B, vCores:0>	<memory:0 B, vCores:0>	<memory:0 B, vCores:0>	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdo
0	0	0	0	1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Sche
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs	% of Queue	% of Cluster	Progress
application_1696092114330_0004	kidusmt	select count(*) from football (Stage-1)	MAPREDUCE	default	0	Sat Sep 30 16:58:12 -0500 2023	N/A	Sat Sep 30 16:58:13 -0500 2023	KILLED	KILLED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1696092114330_0003	kidusmt	select count(*) from football (Stage-1)	MAPREDUCE	default	0	Sat Sep 30 16:56:54 -0500 2023	N/A	Sat Sep 30 16:56:59 -0500 2023	KILLED	KILLED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1696092114330_0002	kidusmt	select count(*) from football (Stage-1)	MAPREDUCE	default	0	Sat Sep 30 16:56:33 -0500 2023	N/A	Sat Sep 30 16:56:39 -0500 2023	KILLED	KILLED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	
application_1696092114330_0001	kidusmt	select count(*) from football (Stage-1)	MAPREDUCE	default	0	Sat Sep 30 16:55:44 -0500 2023	N/A	Sat Sep 30 16:56:21 -0500 2023	KILLED	KILLED	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0	0.0	

Showing 1 to 4 of 4 entries

Thank you