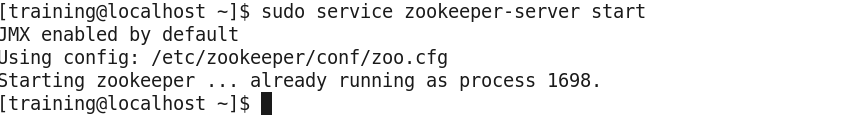
**Assignment 3**

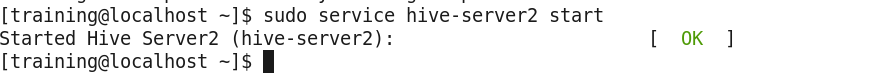
**Chapter 6 Impala Hive**

Setup directory



1. Starting hive server

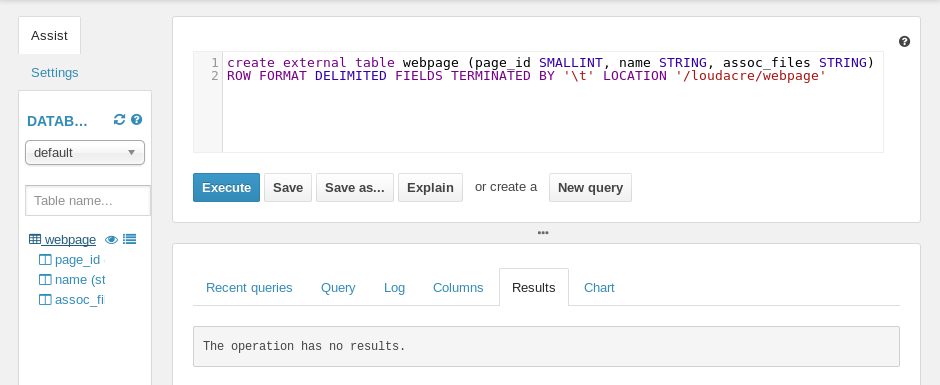




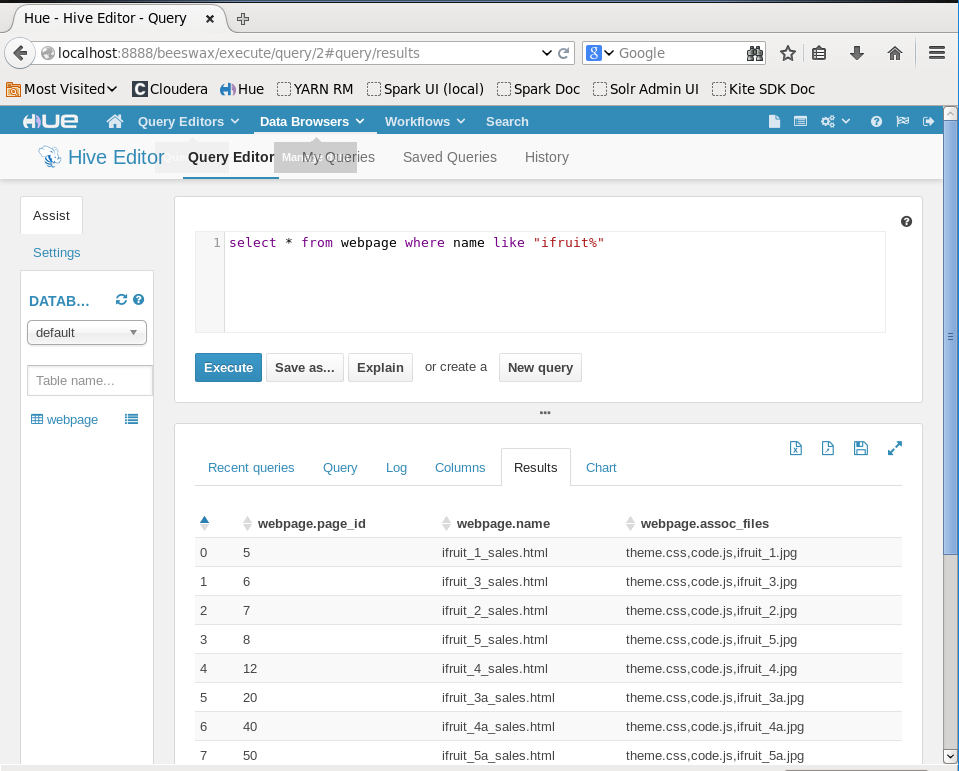
2) 3) 4) 5) 6) Using query editor in hue to execute command.



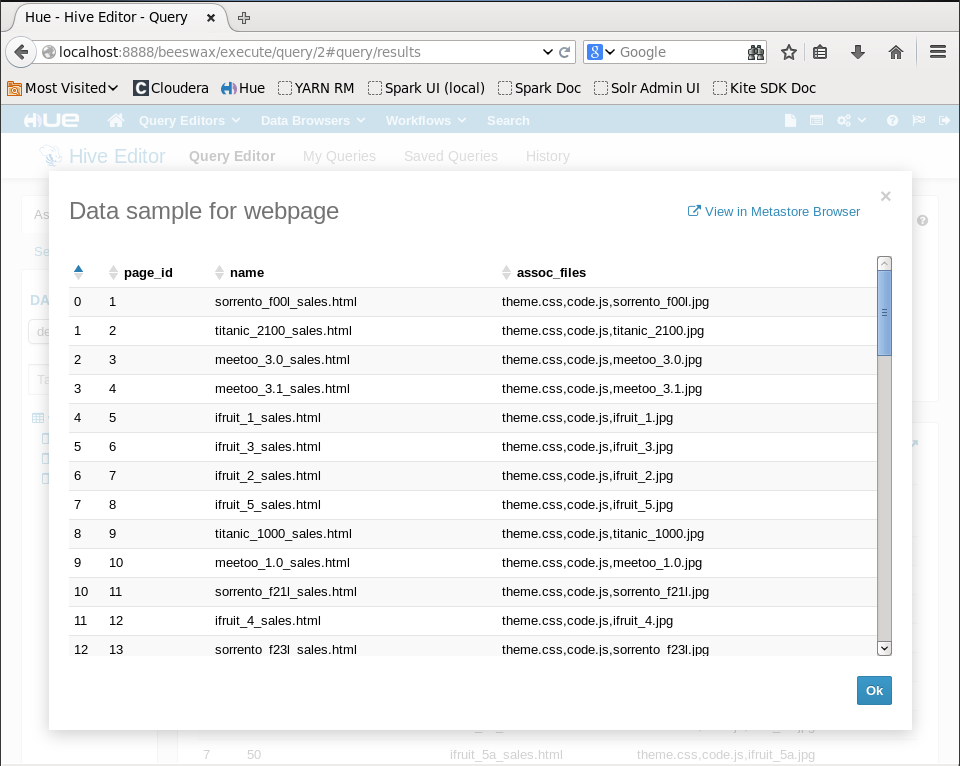
7) Viewing column definition of webpage table.



8) Executing query.

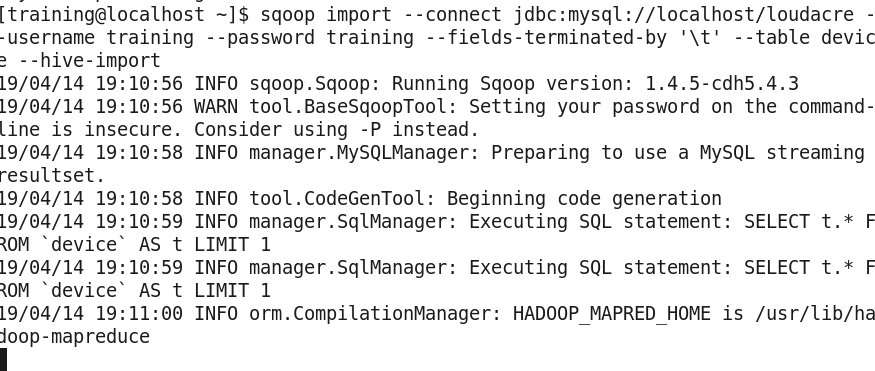


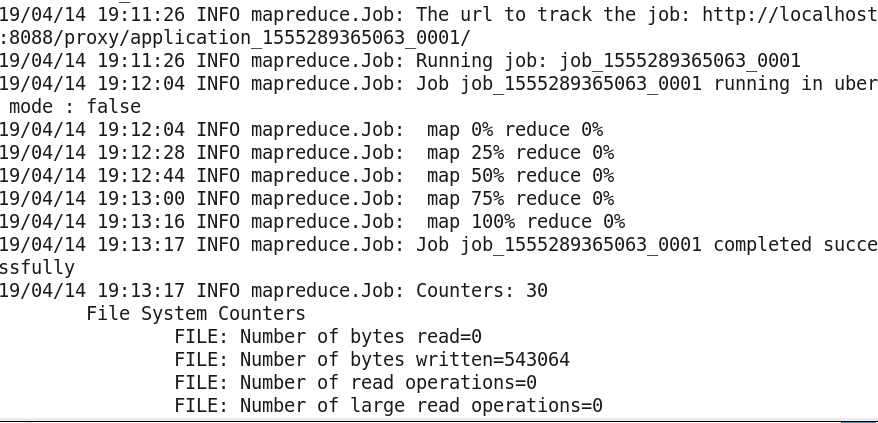
9) Viewing sampling data using preview sample data icon.

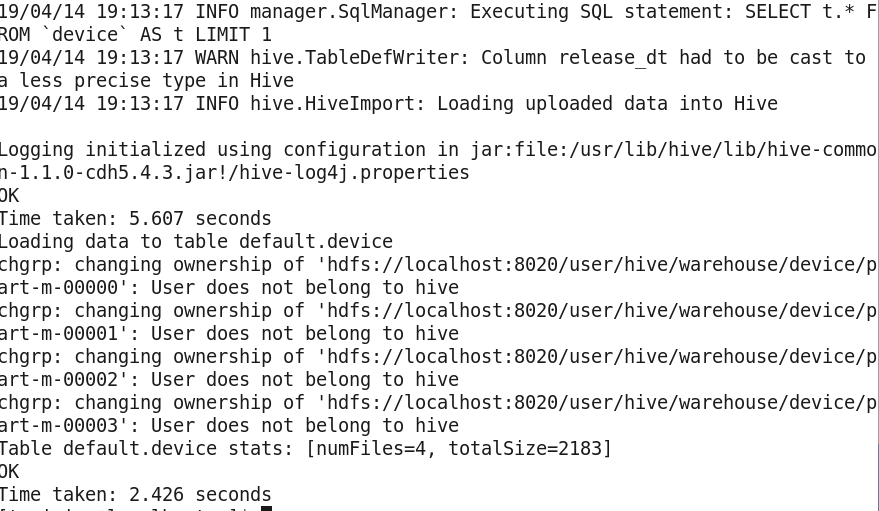


**Use Sqoop to Import Directly into Hive and Impala**

10) Sqoop connection to import data from MySql to HDFS.





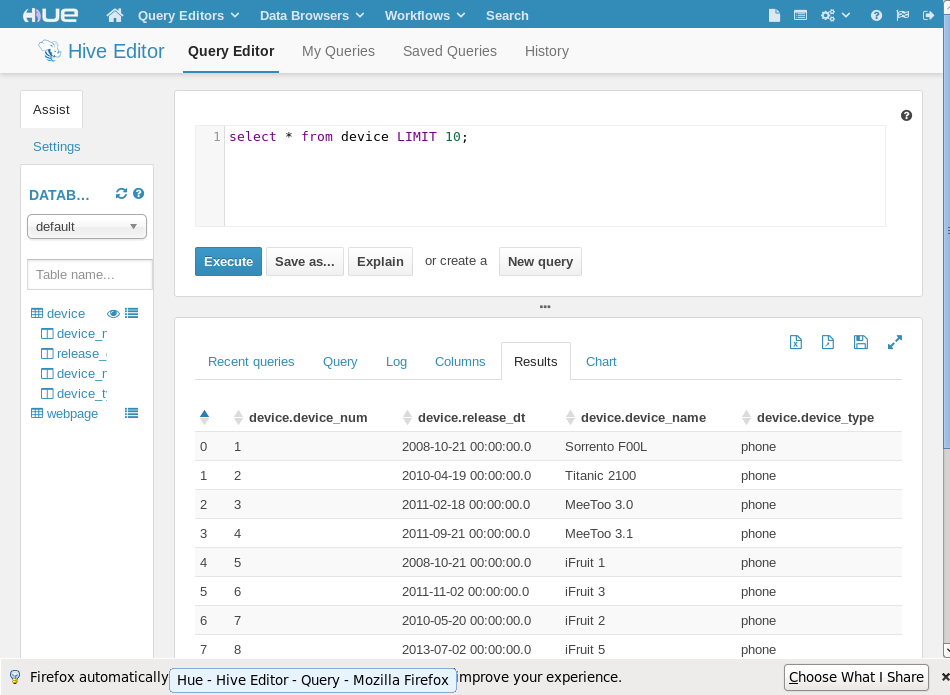


11)Reviewing the imported data.



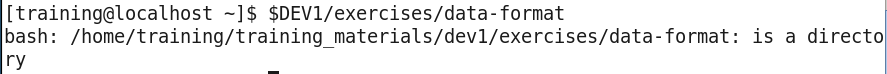
12) Invalidate metadata is used to refresh impala meta data cache. Hive is used here.

13) Executing a test query.



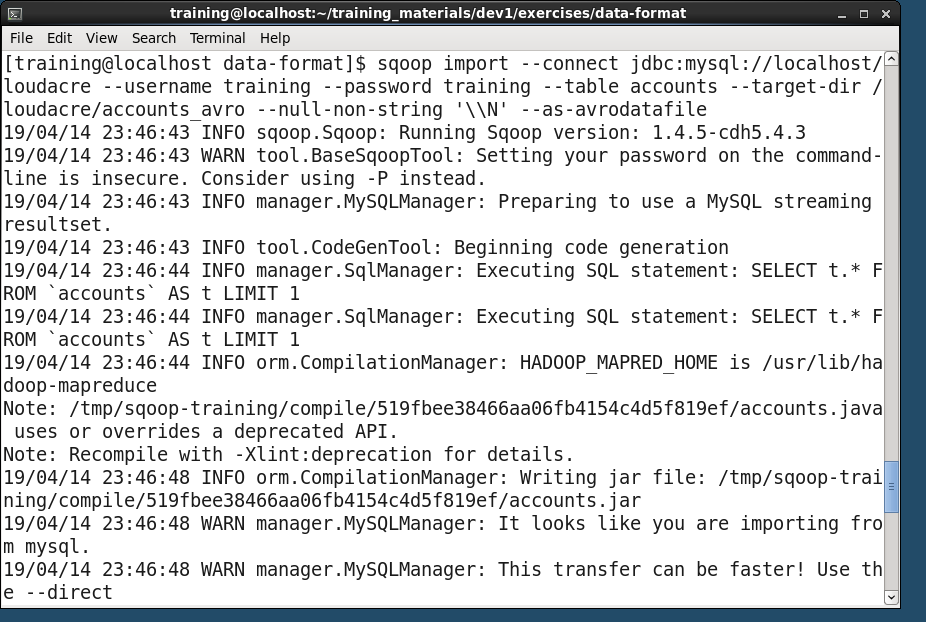
**Chapter 7 File Format**

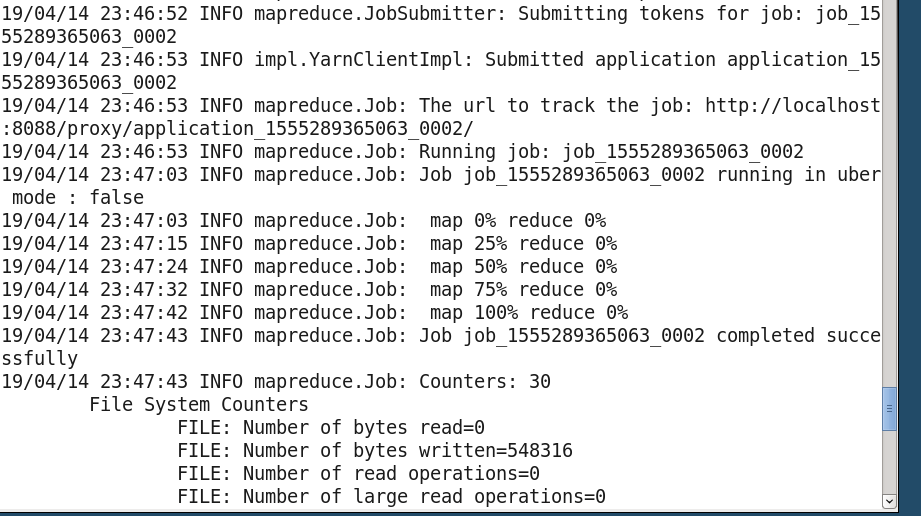
1. Changing to exercise directory.

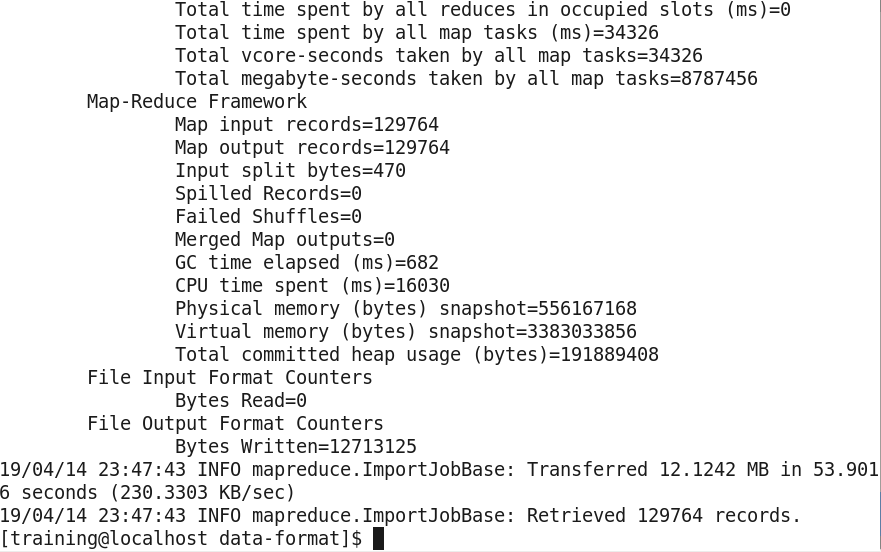




1. Importing the accounts table to an Avro Data Format.







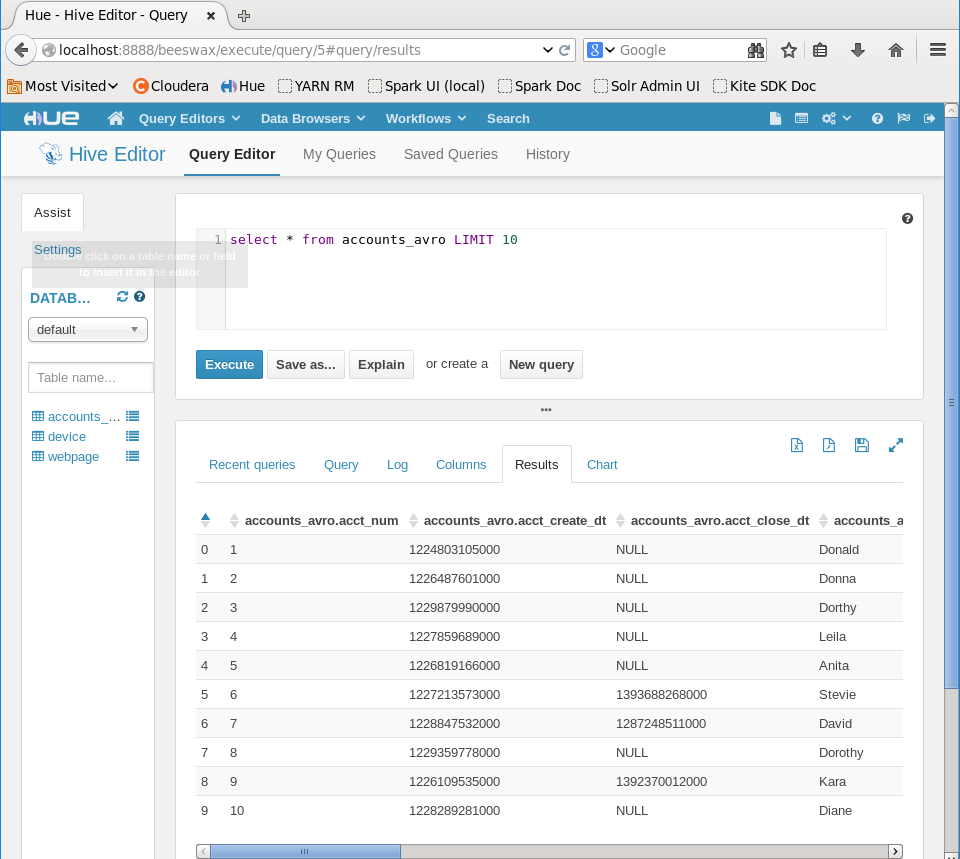
1. Viewing the imported files. Seeing 6 files.

4) Reviewing .avsc file and copy to /loudacre directory.

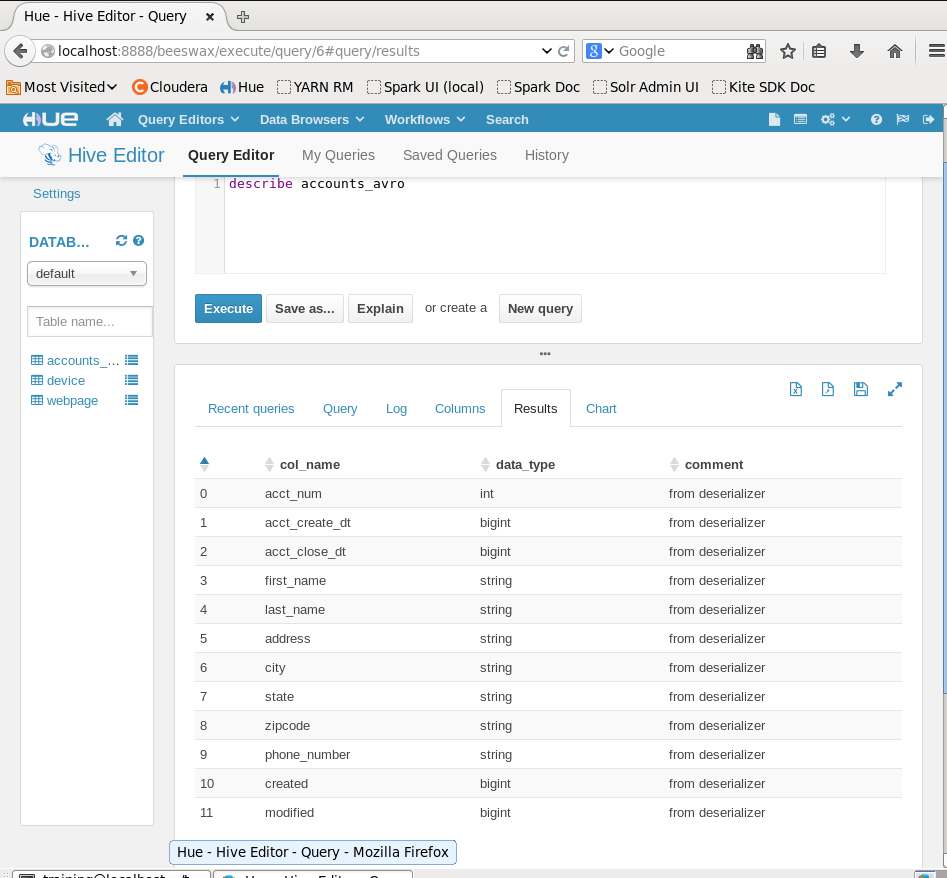


5) Creating a table with a schema.

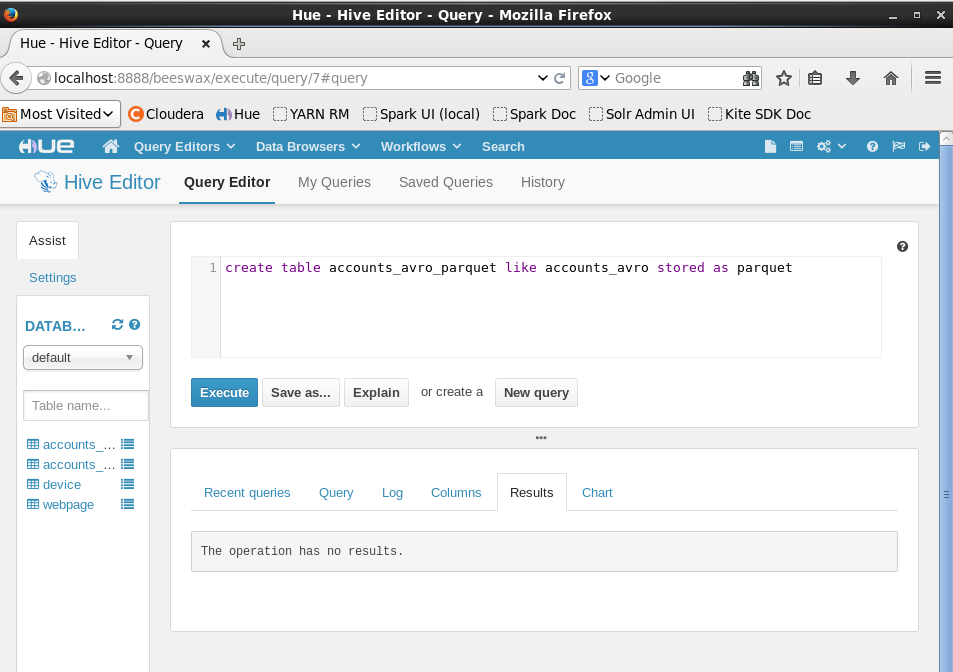
6) Verify creation of the table.



7) Using describe to view the columns and data type.

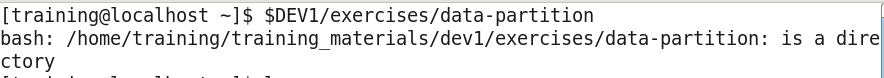


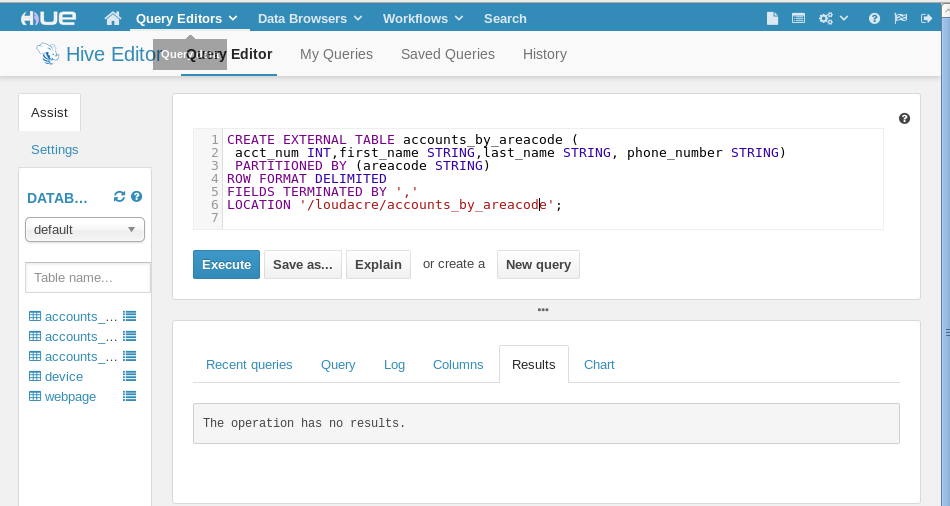
8) Creating a table and storing in parquet format.

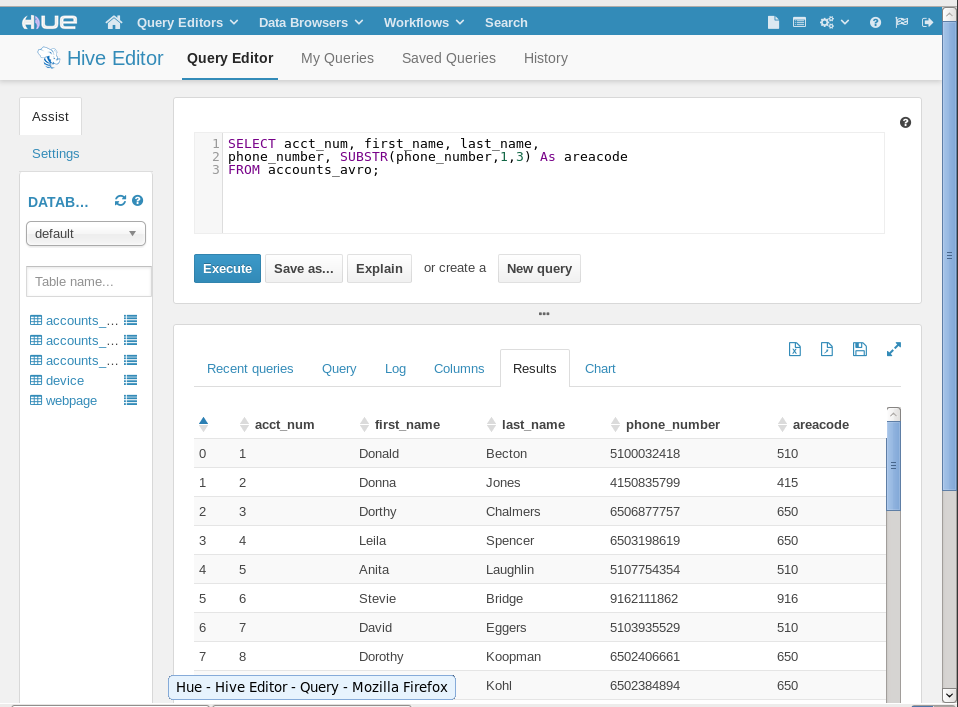


**CHAPTER 8 Partitions**

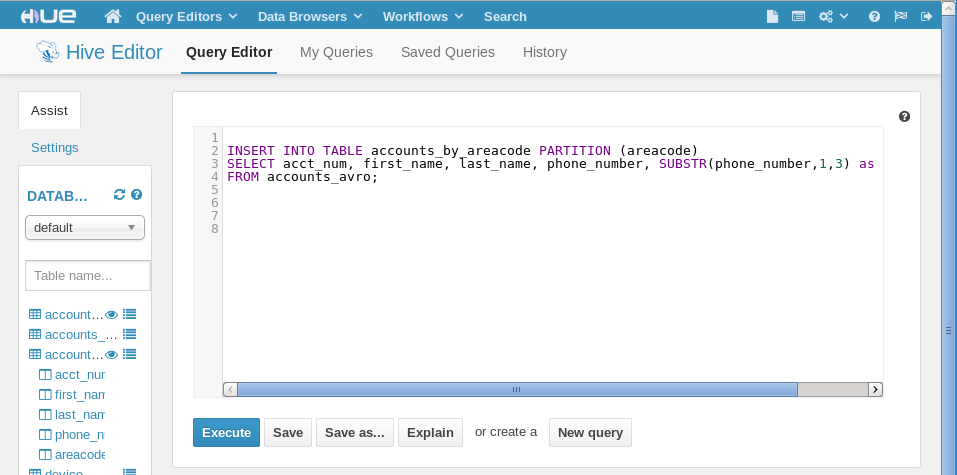
1)Creating an new empty table in hive.



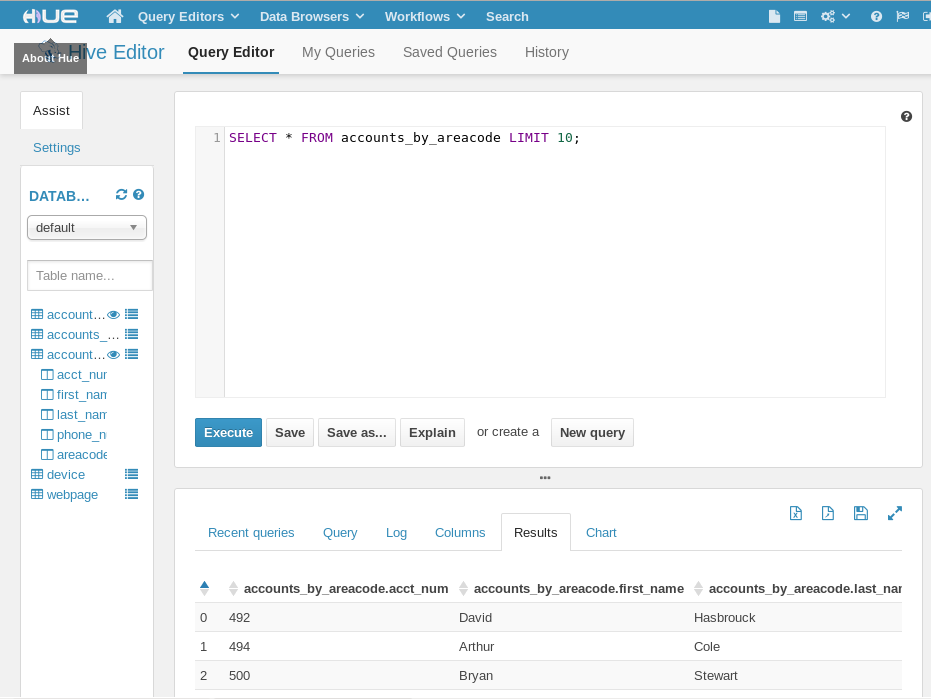


2) Extracting area code from phone number to populate the new table.

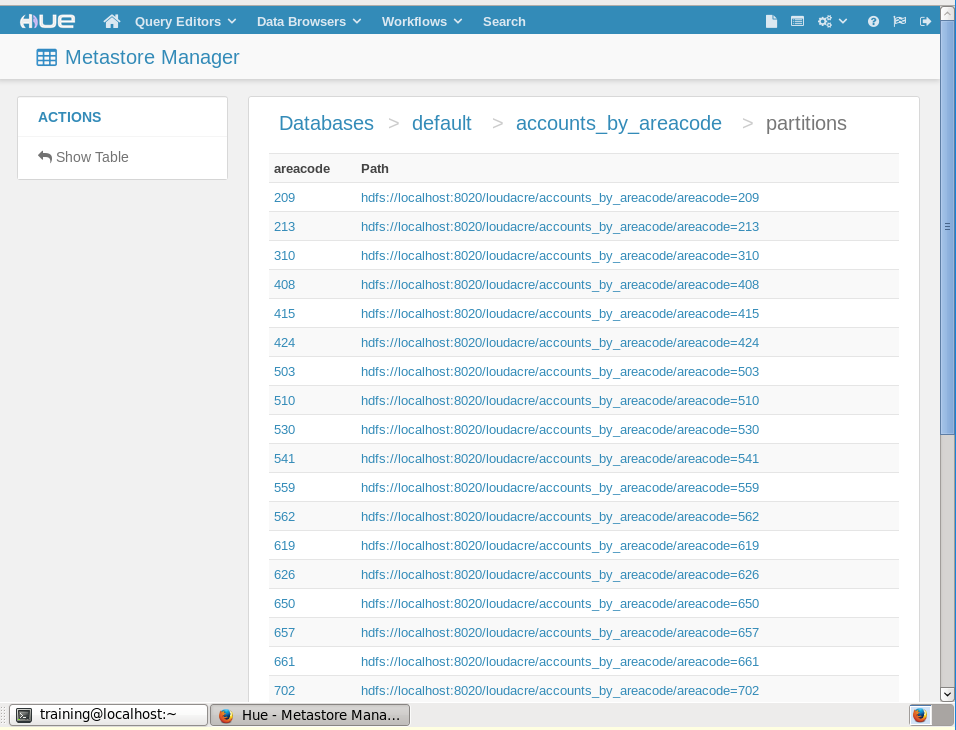
3) Copying data using select and insert command.



4) Executing a simple query to verify.

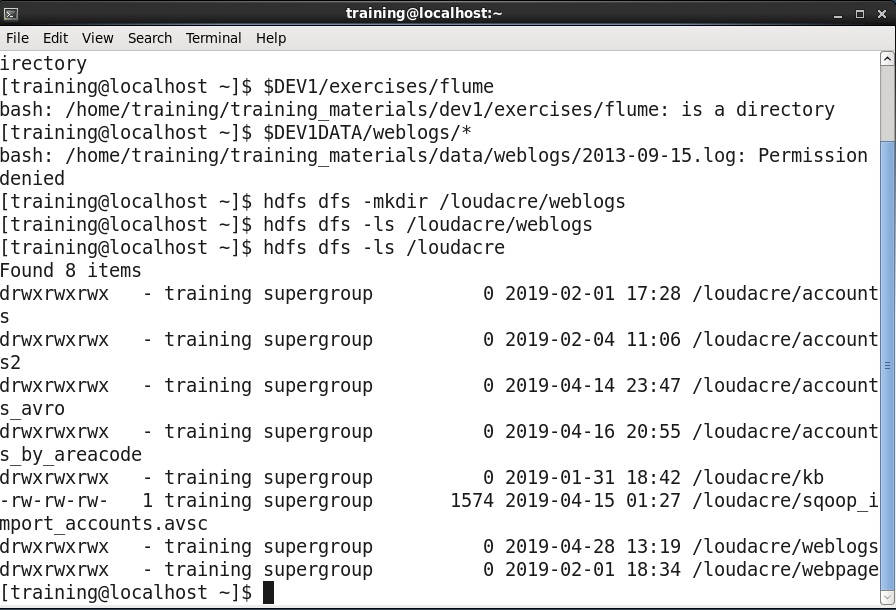


5) Using Hue to confirm directory structure.



**Chapter 9**

1. Creating a directory in HDFS called /loudacre/weblogs to hold fume ingested files.



1. Creating the spool directory into which our web log simulator will store data files for Flume to ingest.



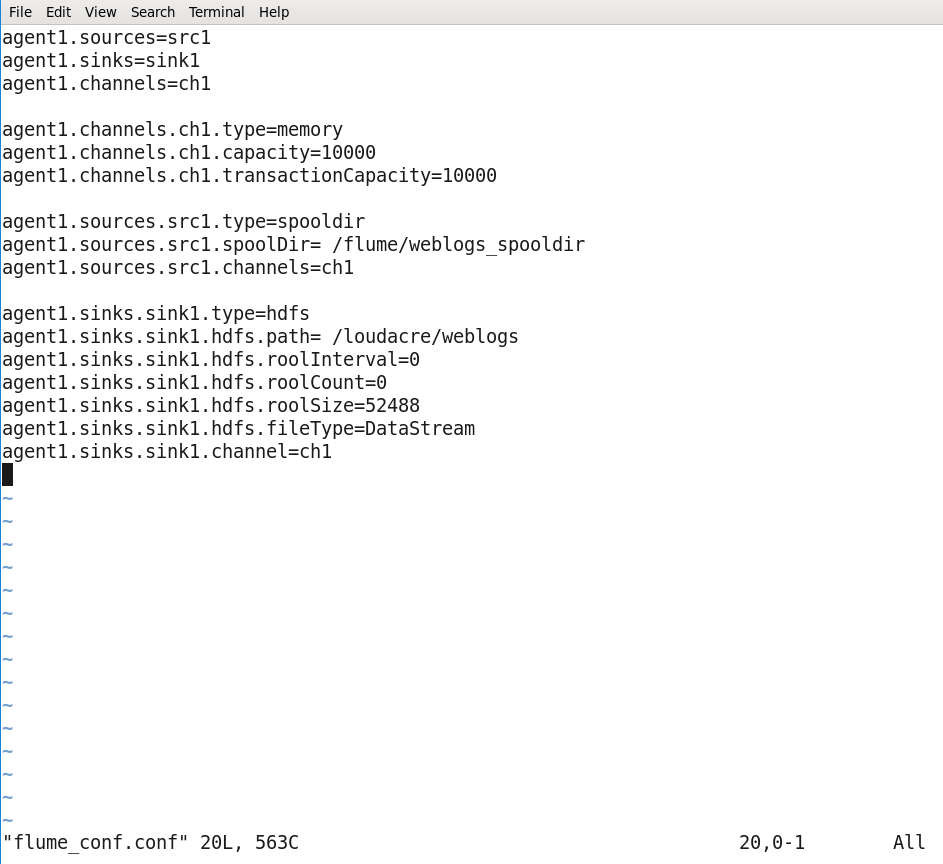
1. Providing all users permission to write in flume/weblogs\_spooldir.

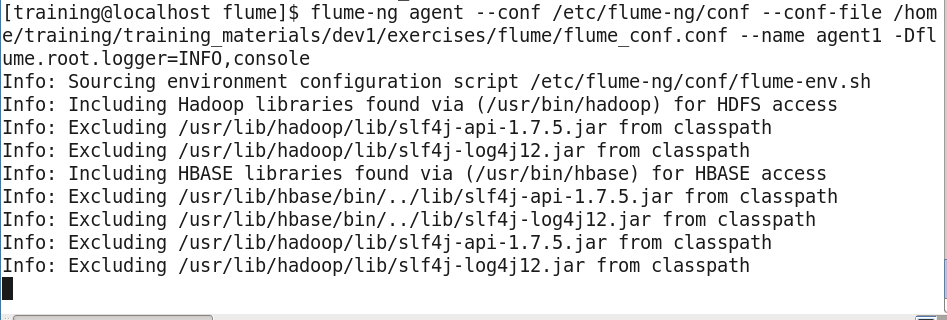


1. Change directory to exercise/flume.

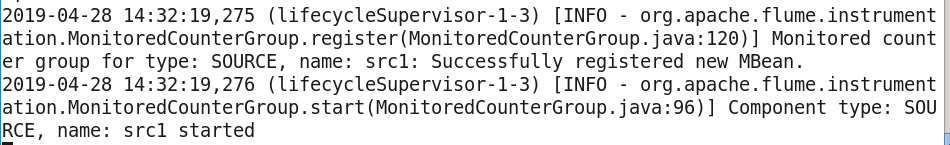


1. Starting flume agent with the configuration.



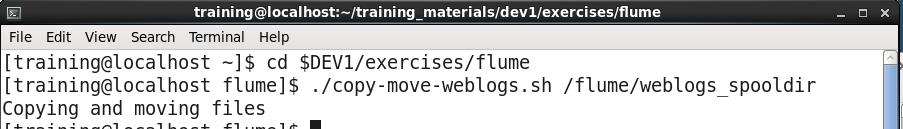


1. Output



Simulate Apache web server output

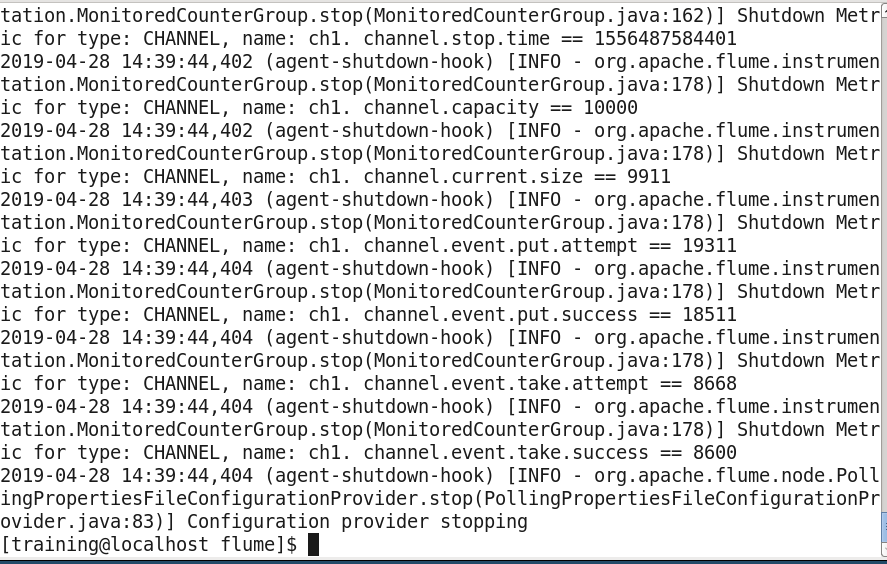
1. Open a separate terminal and copy the weblogs in /flume/weblogs\_spooldir directory.



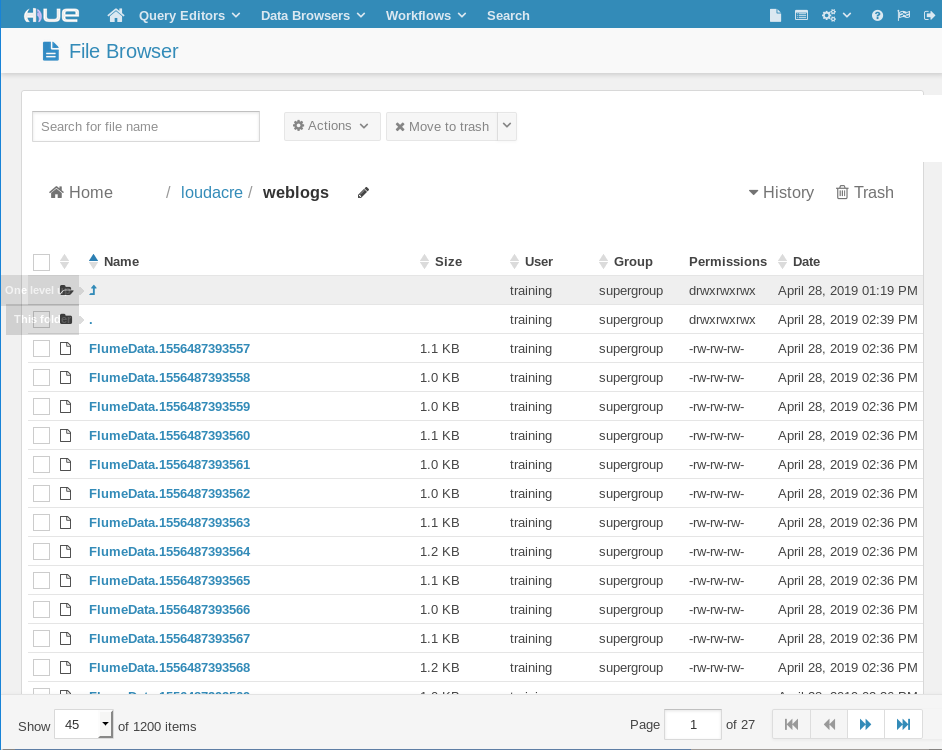
1. Logging outputs



1. Terminate process by using ctrl + c.

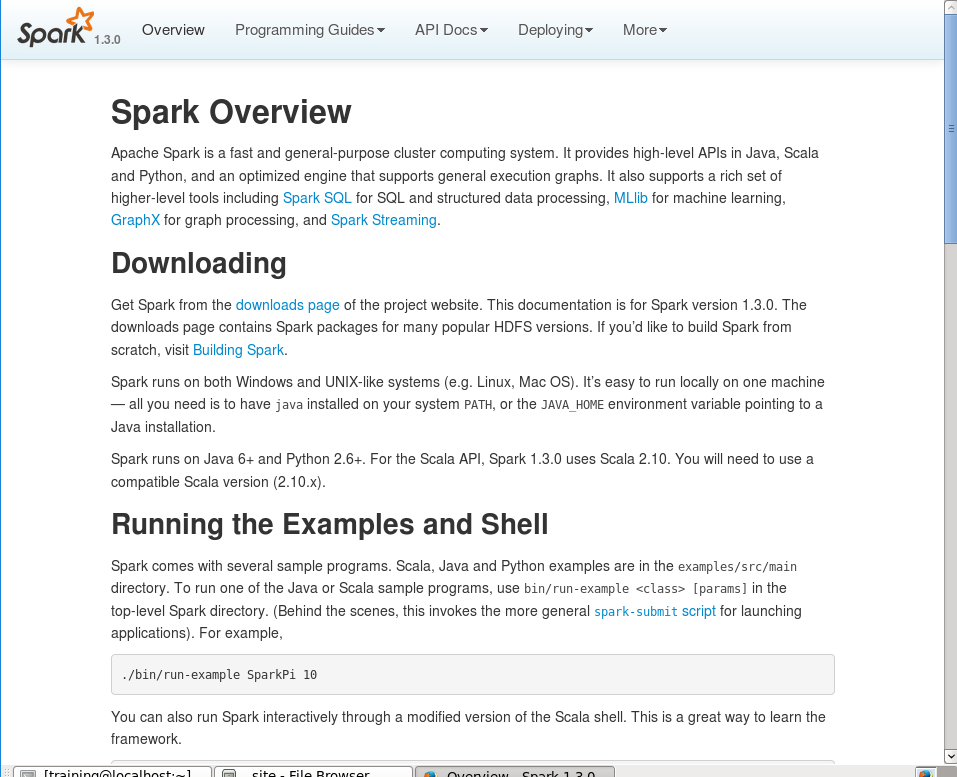


1. Files added by the flume agent.

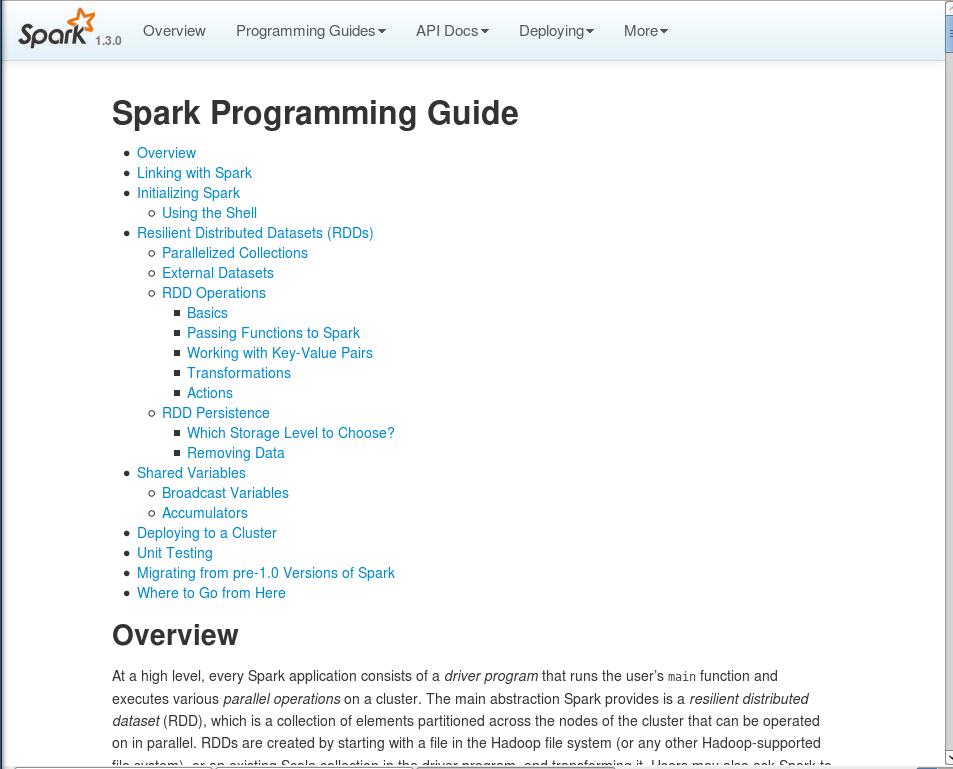


**Chapter 10**

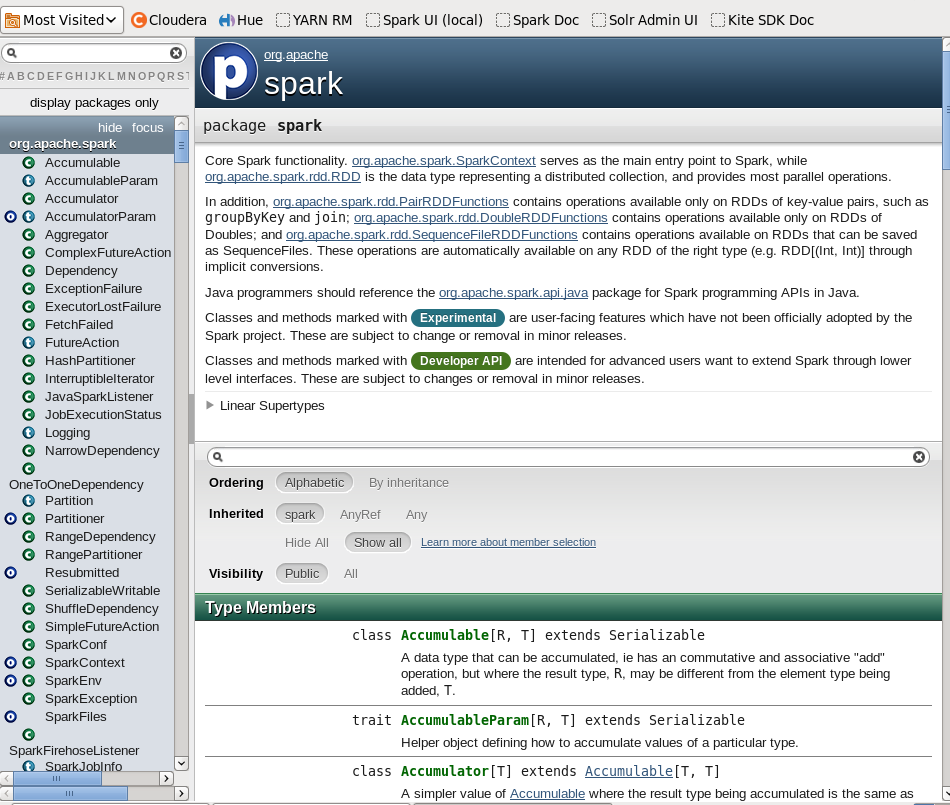
1. Spark documentation



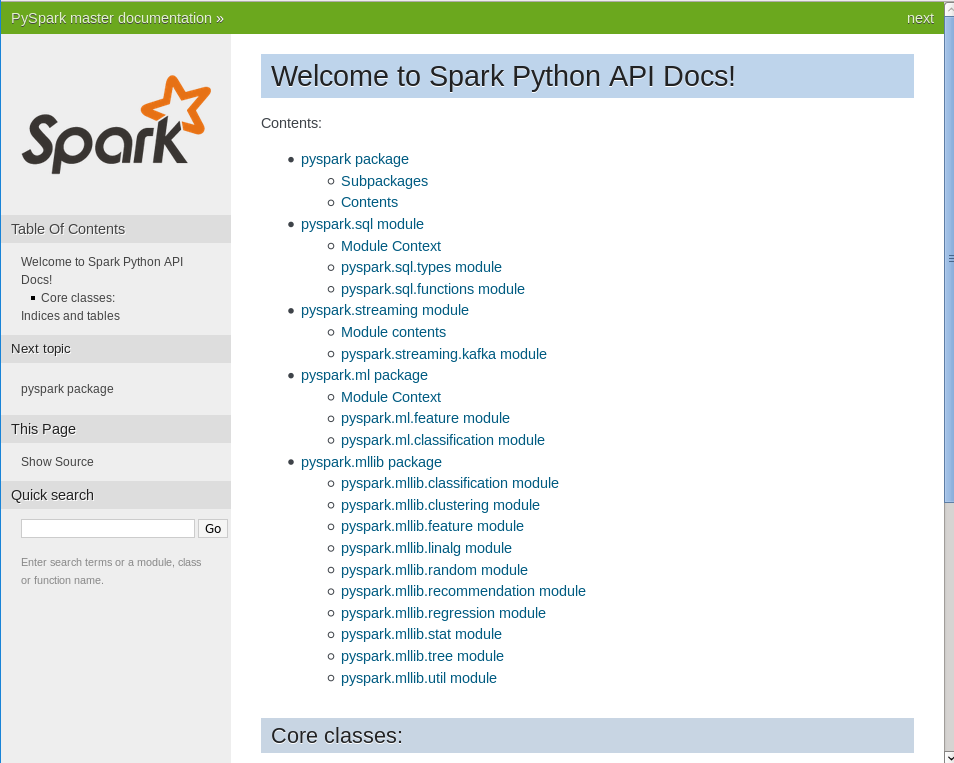
1. Spark Programming Guide.



1. SCALA API

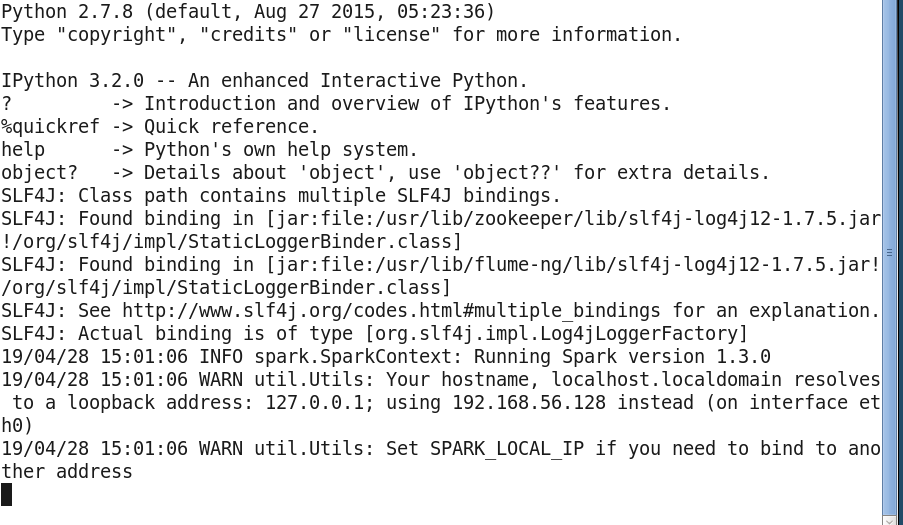


Python API



**Chapter 10 Spark shell RDD**

1. Starting pyspark.



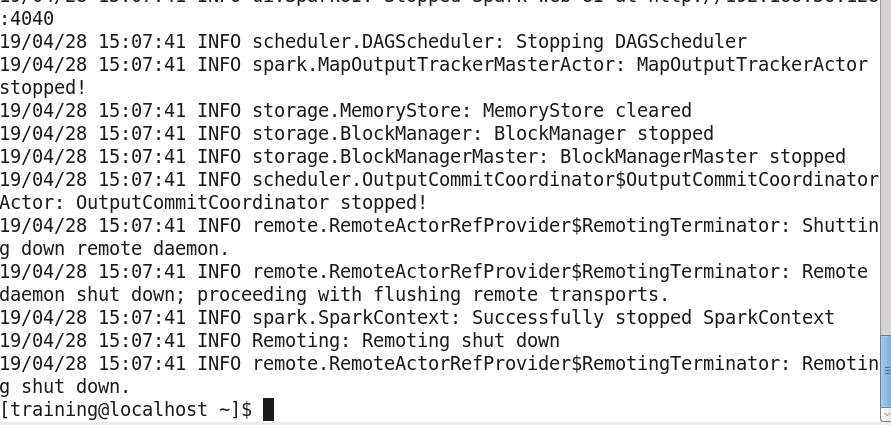
1. Check SparkContext



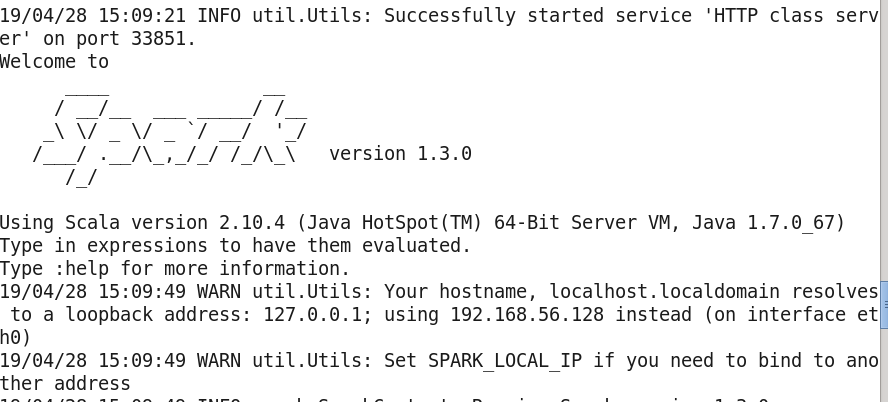
1. Viewing available SparkContext methods.

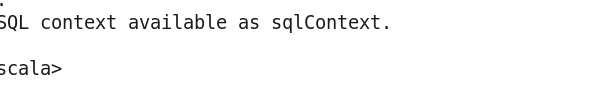


1. Exit shell



1. Starting scala-spark shell.





1. Calling spark context object.

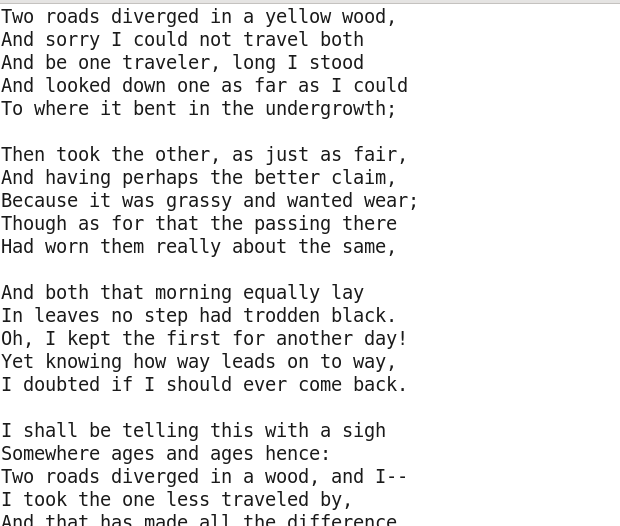


1. View available spark context methods.

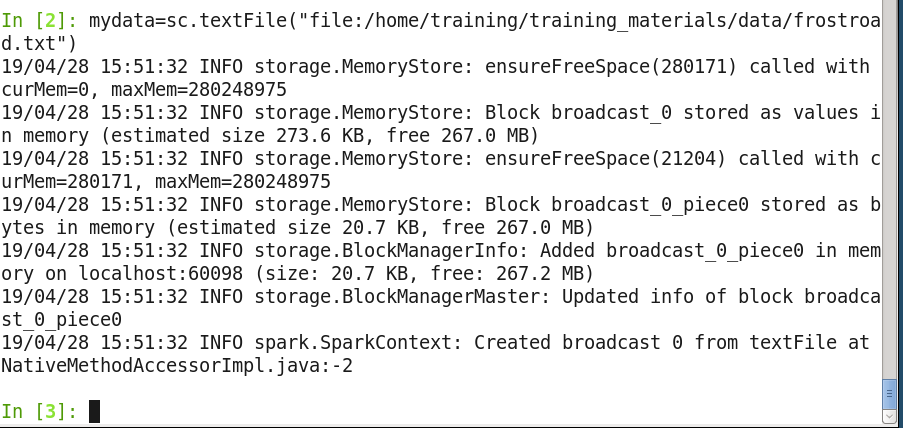


Load and view text file (Python or Spark)

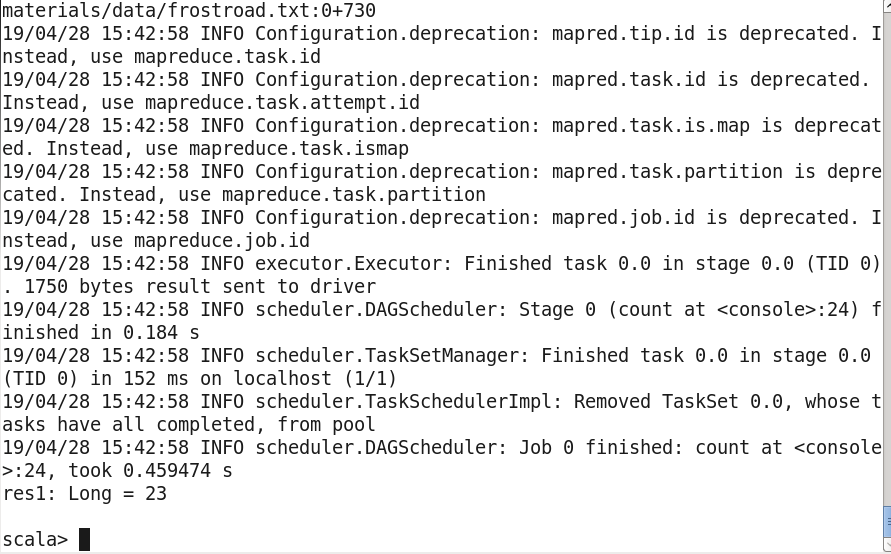
1. Viewing frostroad.txt



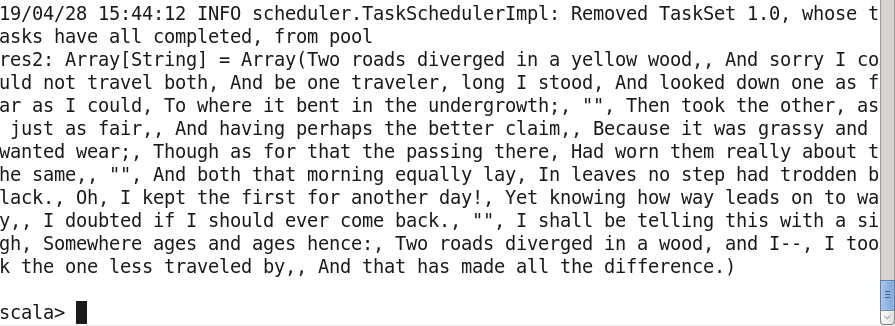
1. Defining an RDD with the data.

**IN PYTHON**

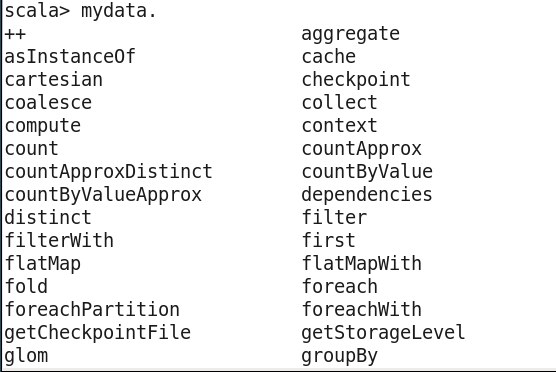
1. Performing count operation to make sure spark reads the file.

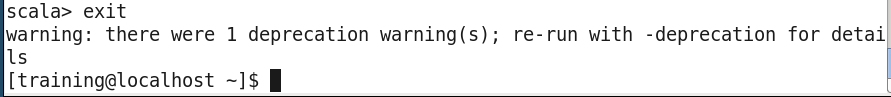


1. Displaying the entire data set using collect operation.



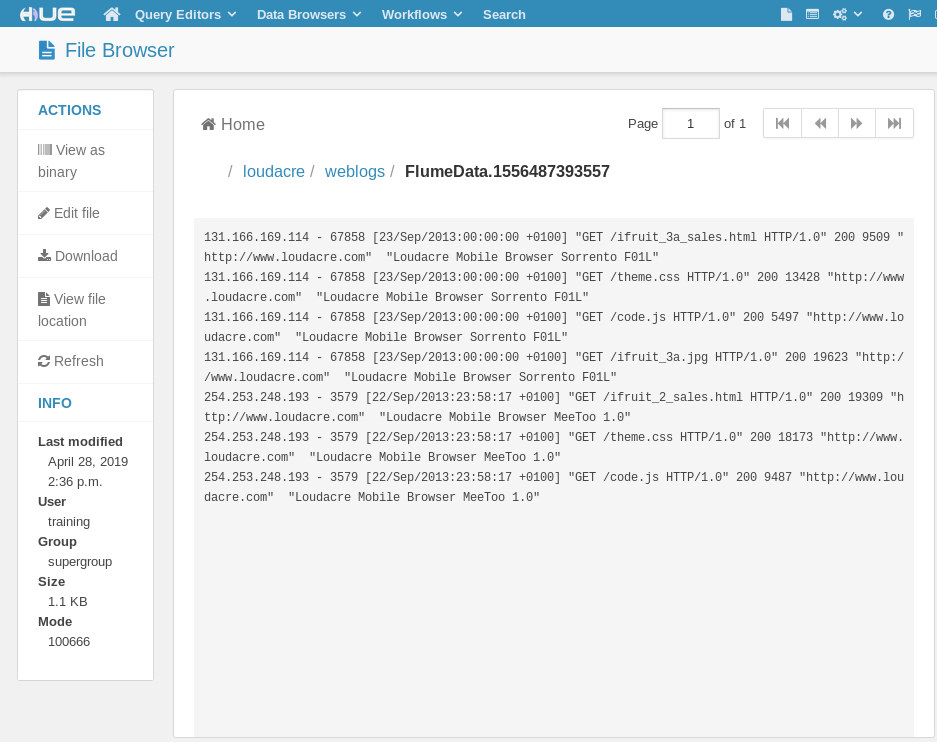
1. Viewing all available transformation and operations that can be performed on an RDD.



1. Exit the shell.

**Chapter 10 Use RDD**

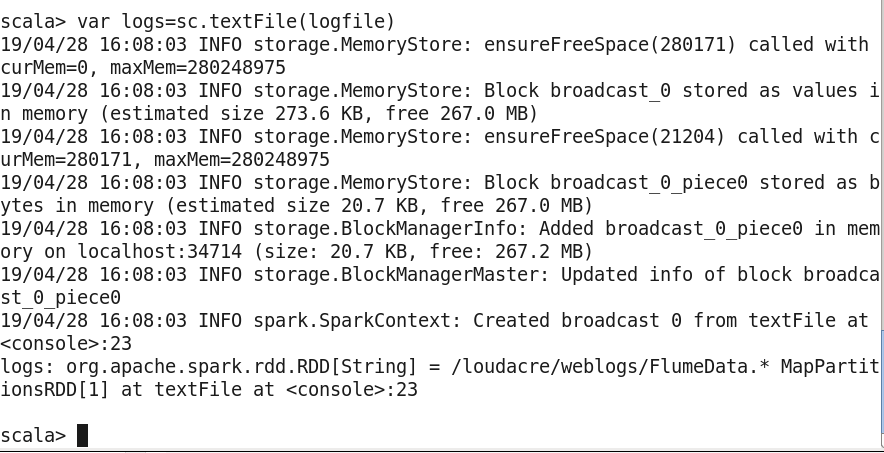
1. Reviewing one of the web server log files captured by fume.



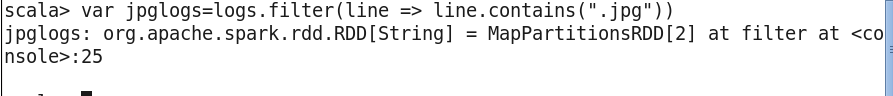
1. Setting a variable for the log file.



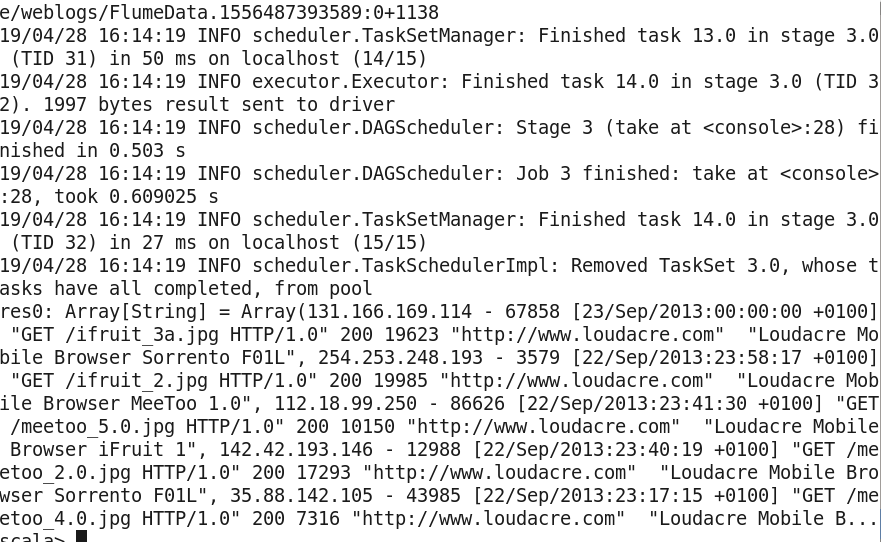
1. Creating an RDD for the data file.



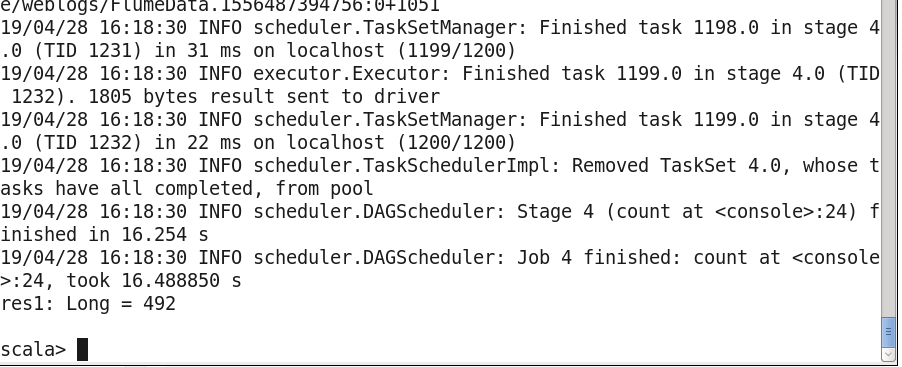
1. Creating an RDD containing only those lines that are requests for JPG files.



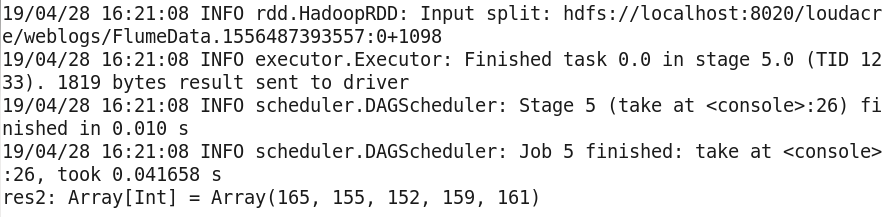
1. Viewing first 10 lines using take.



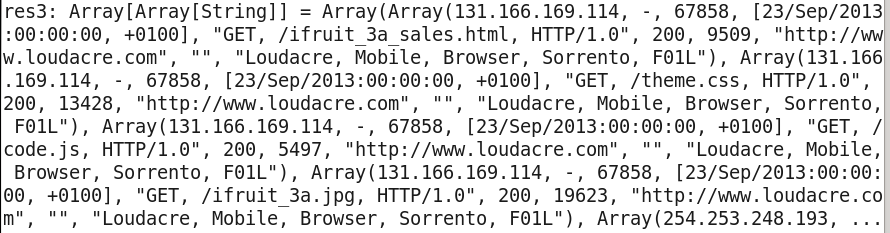
1. Counting the number of JPG requests.



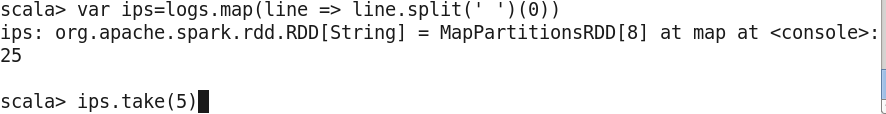
1. Using the map function to define a new RDD and return length of each line in log file.

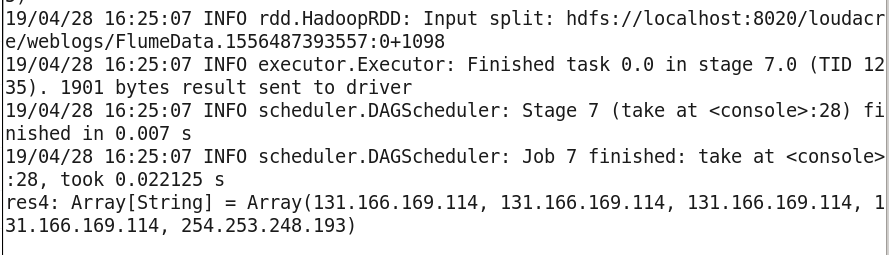


1. Mapping to an array of words for each line.

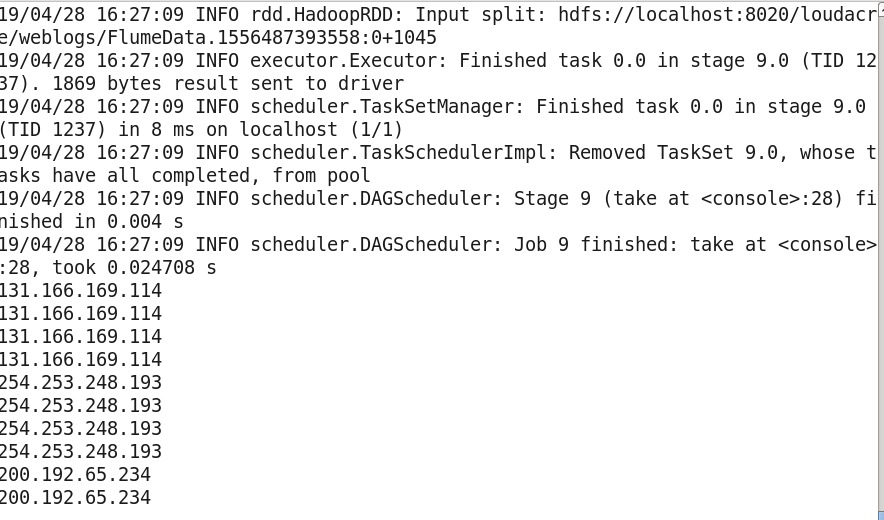


1. Defining a new RDD containing just the IP addresses from each line in the log file.

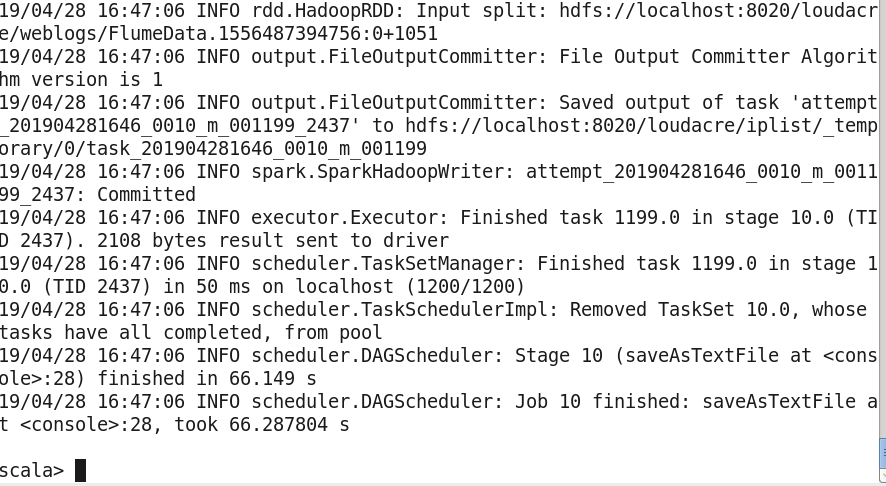




1. Reading through array iteration.



1. Saving IP address as a Text file.



1. Viewing contents of /loudacre/iplist in hue and review the content of one file.

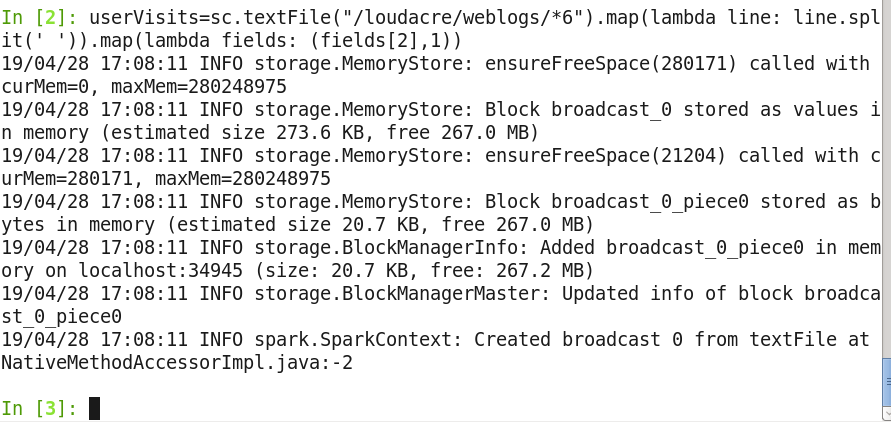


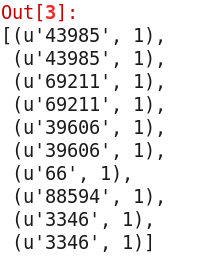
**Chapter 12 Pair RDD**

Explore Web Log Files

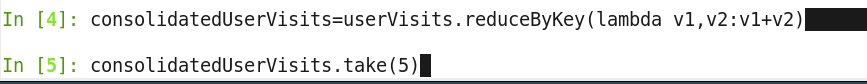
1. A) Using map to create a Pair RDD with the user ID as the key, and the integer 1 as the value.

Using Take to see the key value pairs.



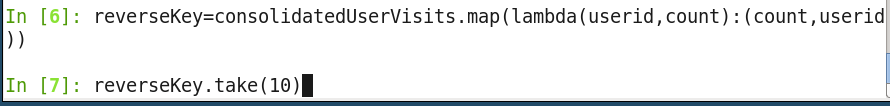


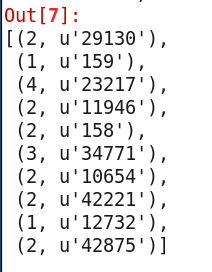
B) Using reduce to sum the values for each user id.





1. A) Using map to reverse the key and value.



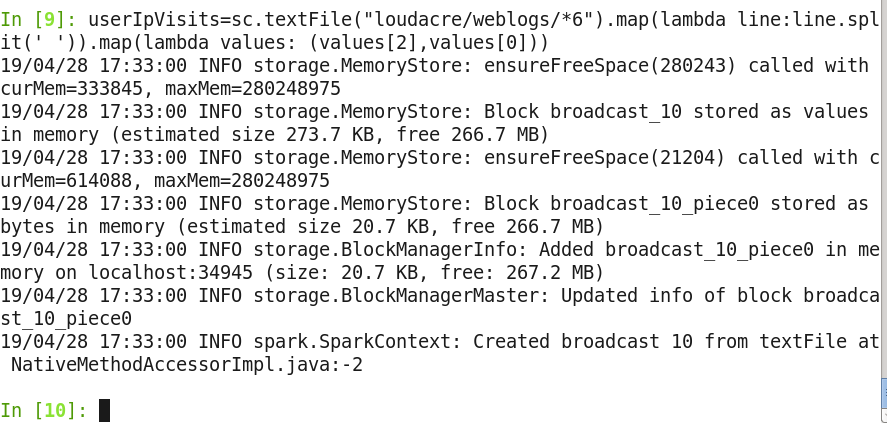


B) Using the countByKey to return a Map of frequency: user -­ count pairs.

reverseKey.countByKey()

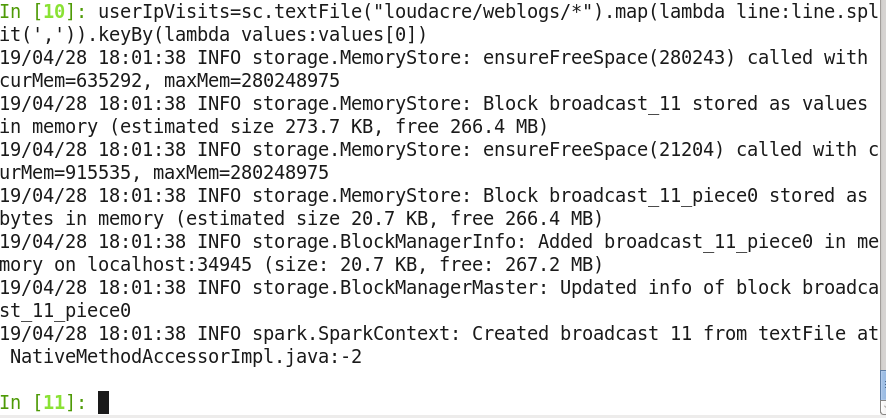


1. Creating an RDD where the user id is the key, and the value is the list of all the IP addresses.

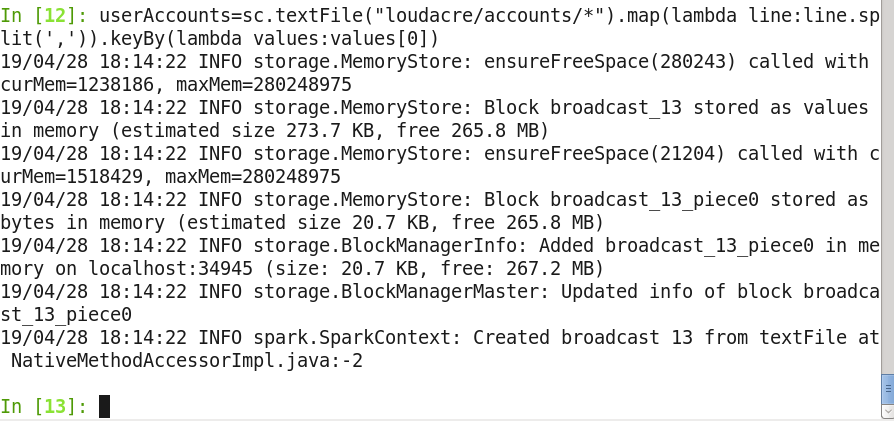


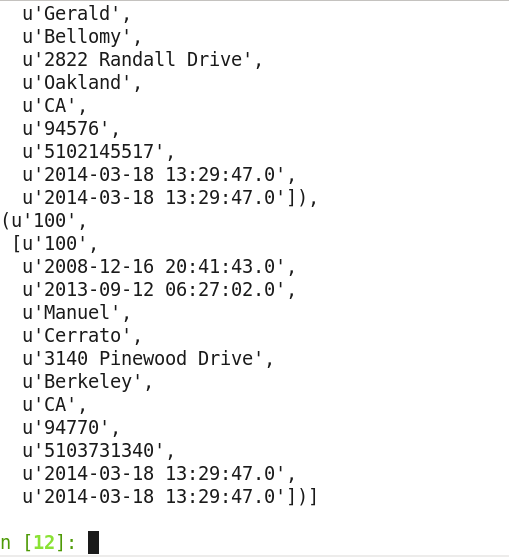
Join Web Log Data with Account Data

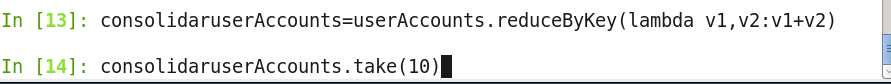
1. A) Creating an RDD based on accounts data consisting of key/value array pair.



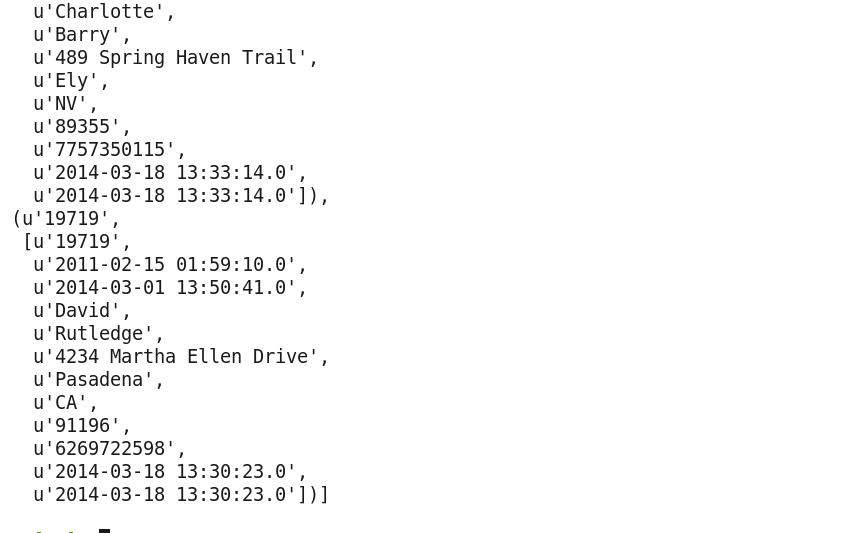
B) Joining the Pair RDD with the set of user-­‐id/hit-­‐count pairs.





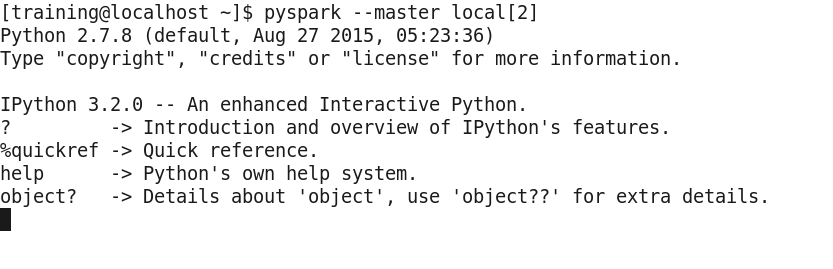


C)

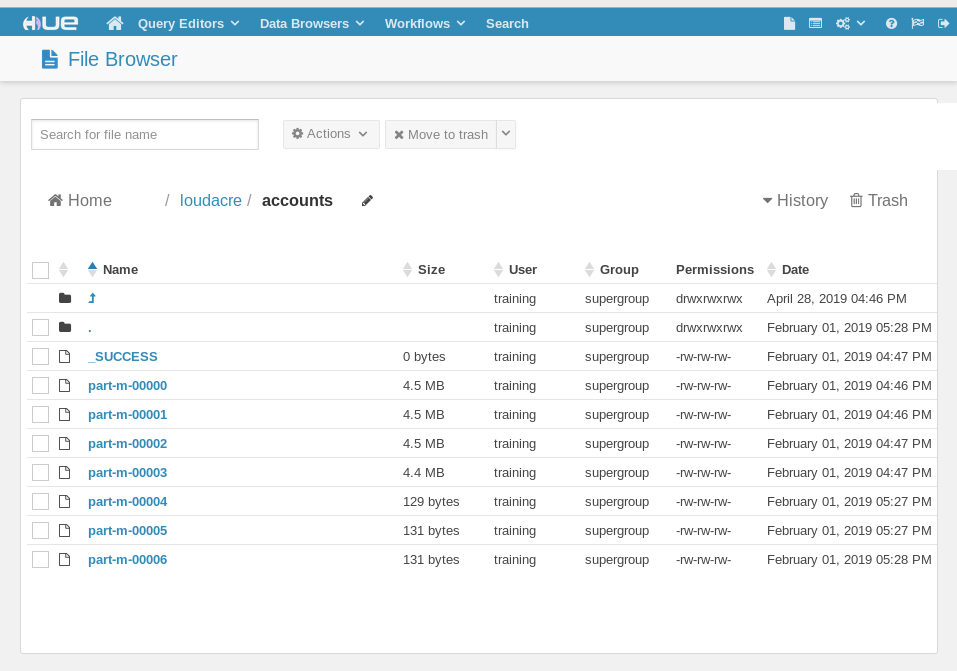


**Chapter 14 View Jobs**

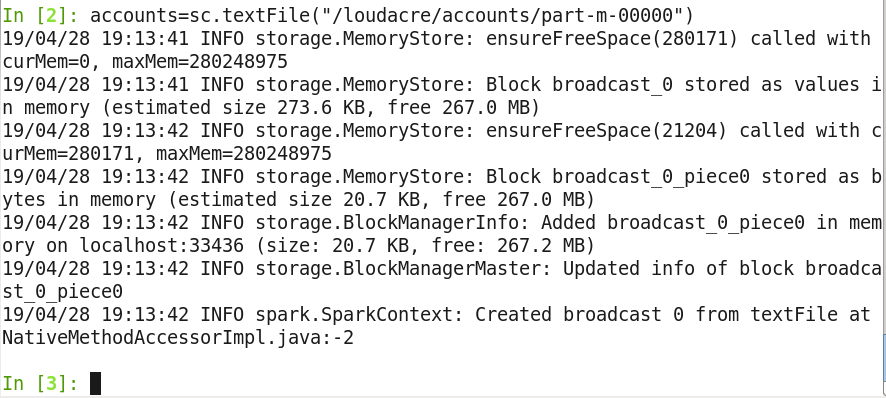
1) Spark application with 2 threads.

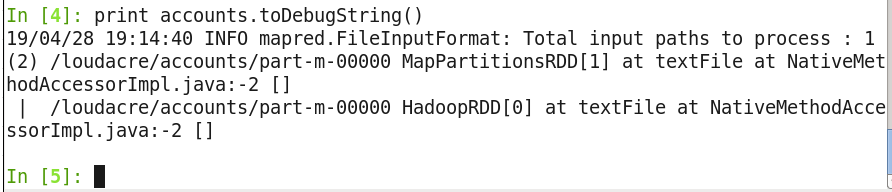


2) View directory in hue.

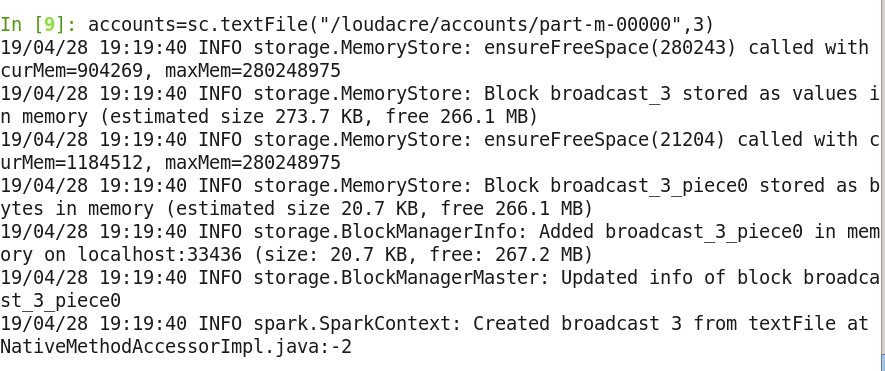


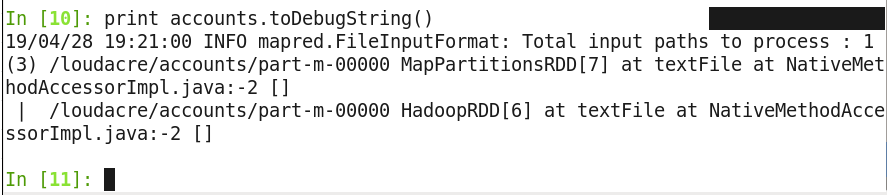
1. Creating RDD based on single file in data set.





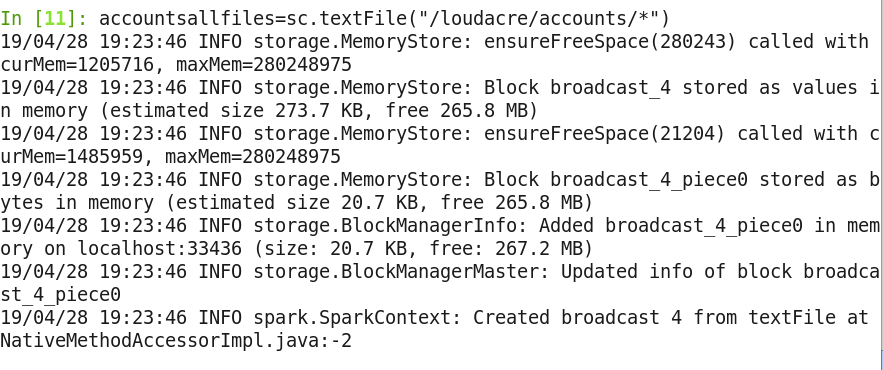
1. Repeating with 3 partitions.





RDD have correctly 3 partitions.

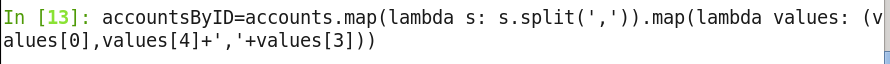
1. RDD for all files.



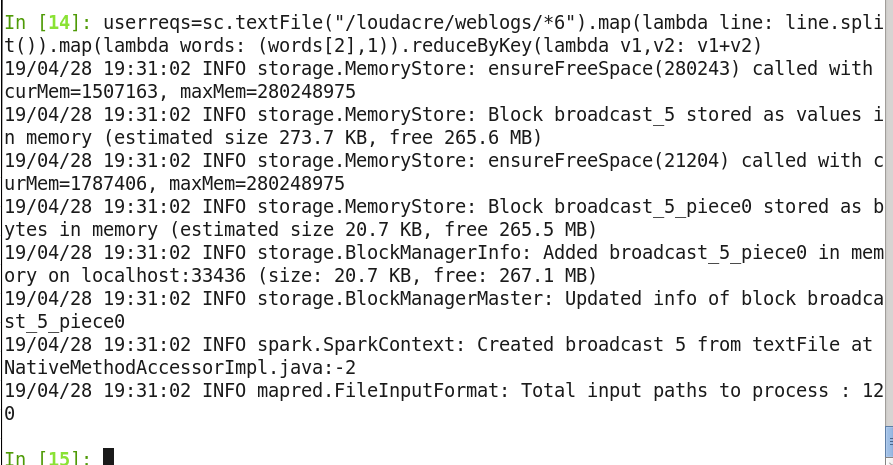


**SET UP the JOB**

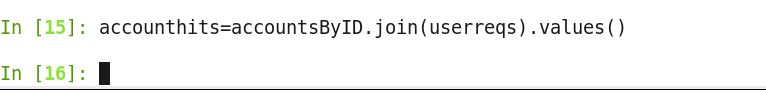
1. creating an RDD of accounts, keyed by ID and with first name, last name for the value.



1. Constructing userreqs RDD



1. Join the 2 RDD by user ID.

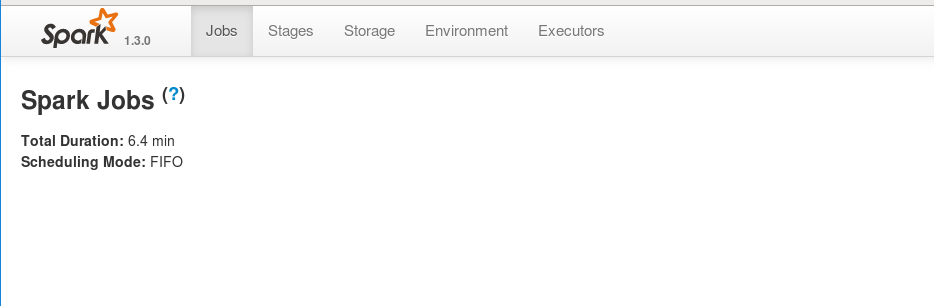


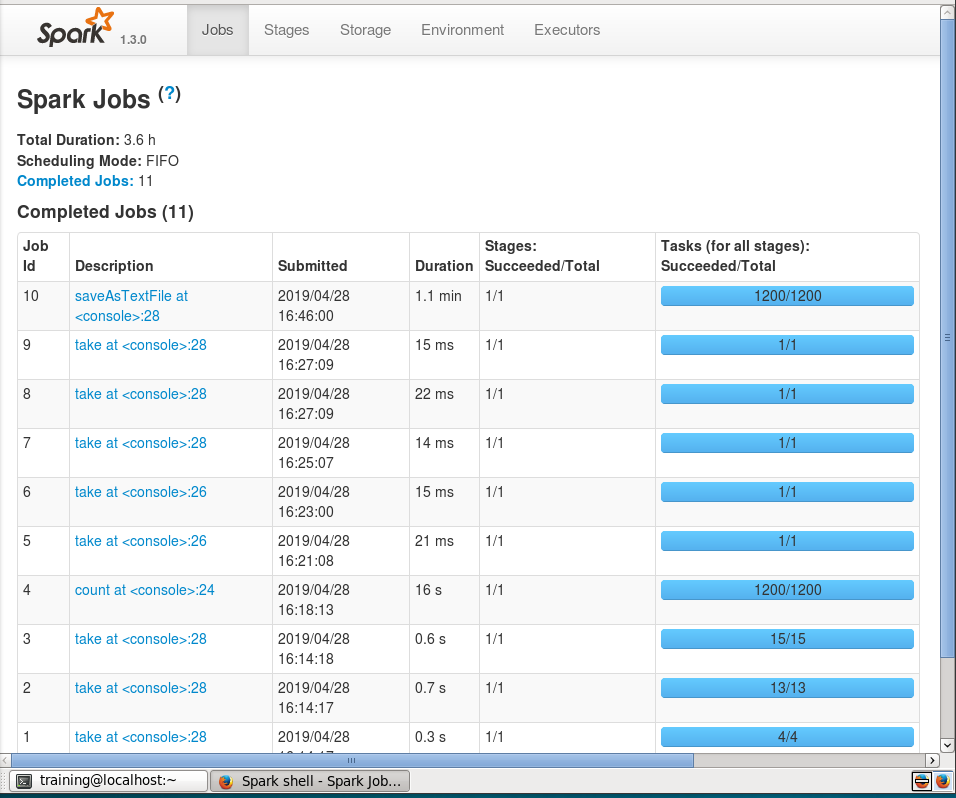
1. Review Output.

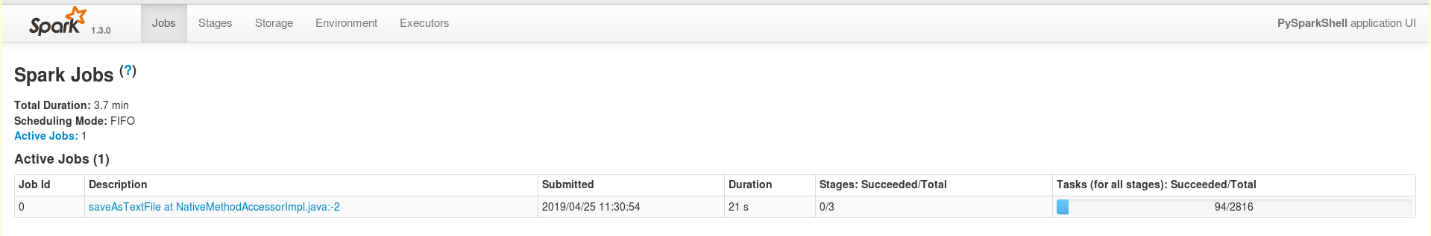
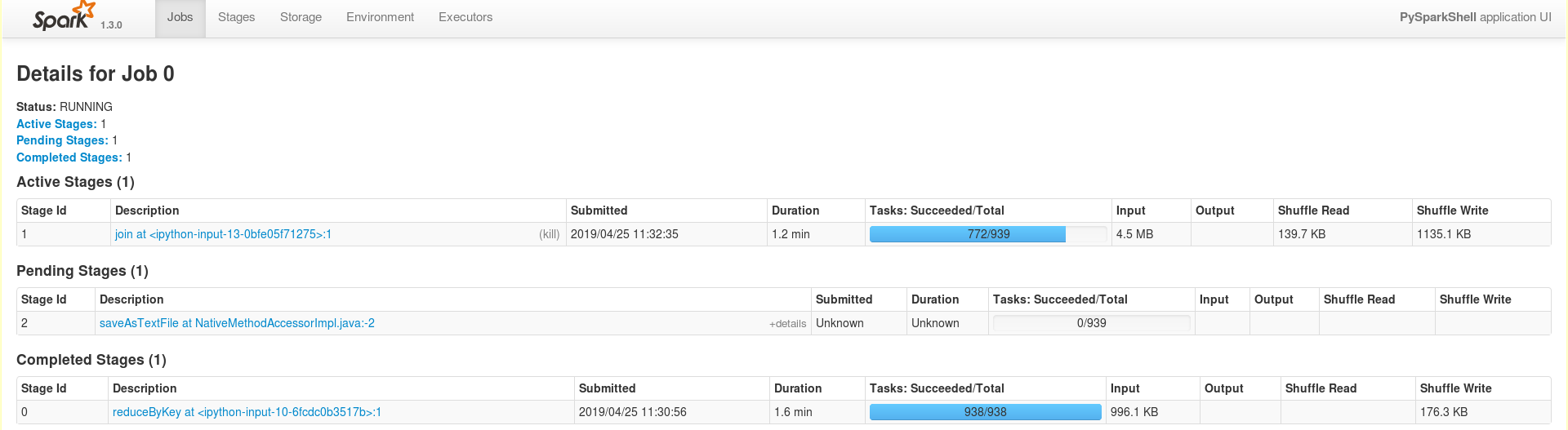


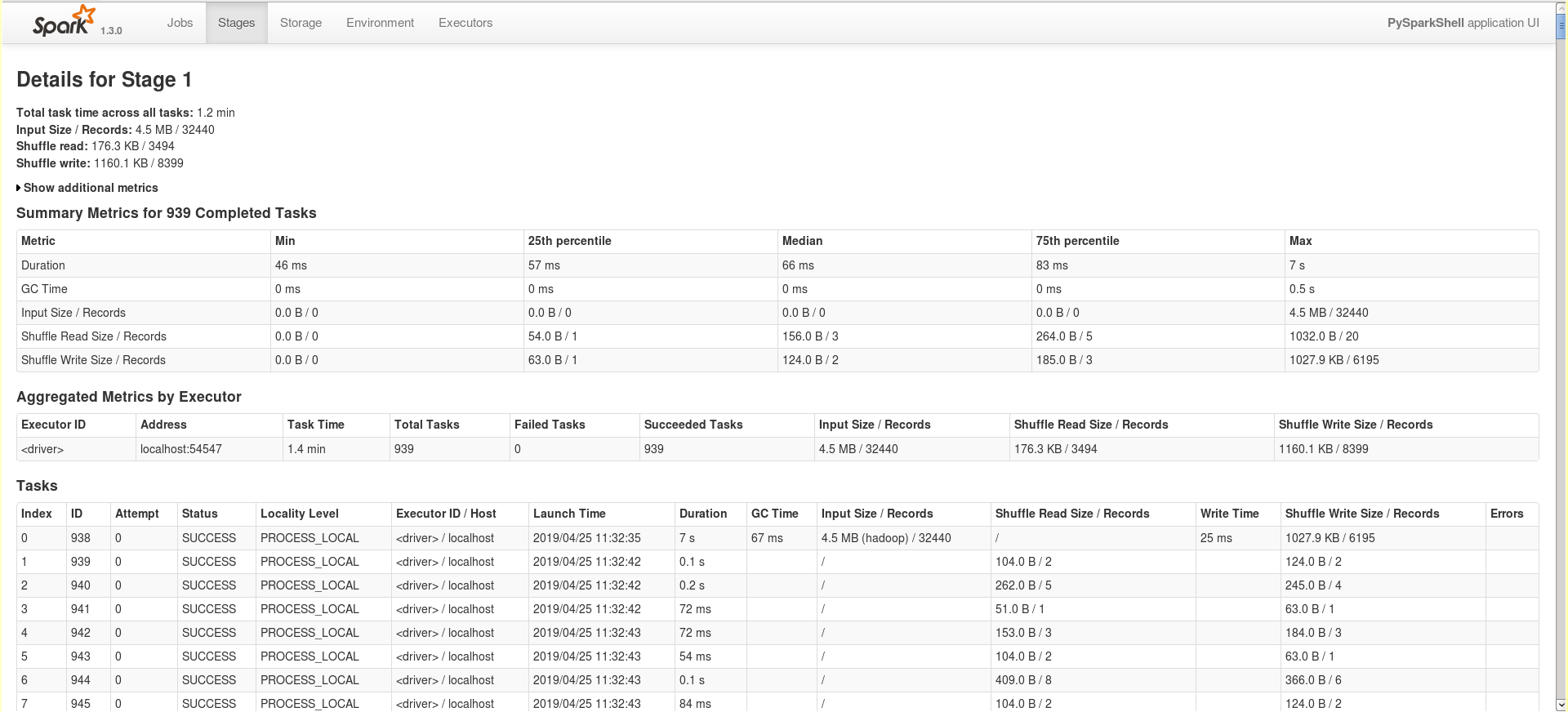
Run the job and review it in the Spark Application UI

1. Spark application.



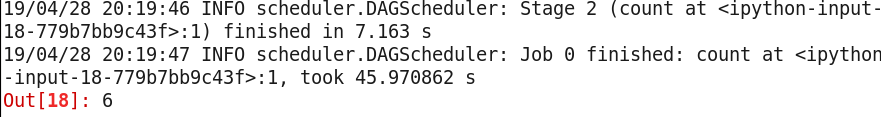


1. Running saveastext.
2. 
3. Job Description details.



**Chapter 15 Persist RDD**

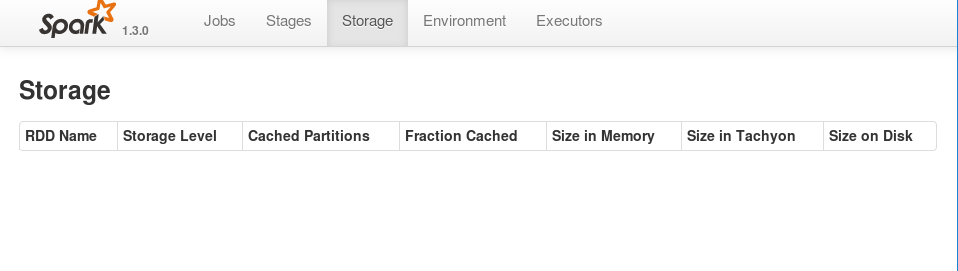
1. Run Pyspark
2. Counting the number of user accounts with a total hit count greater than five.



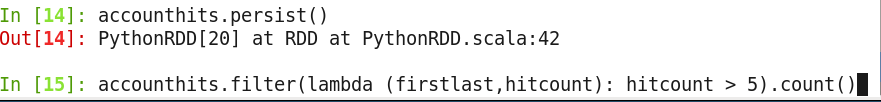
1. Caching the RDD.



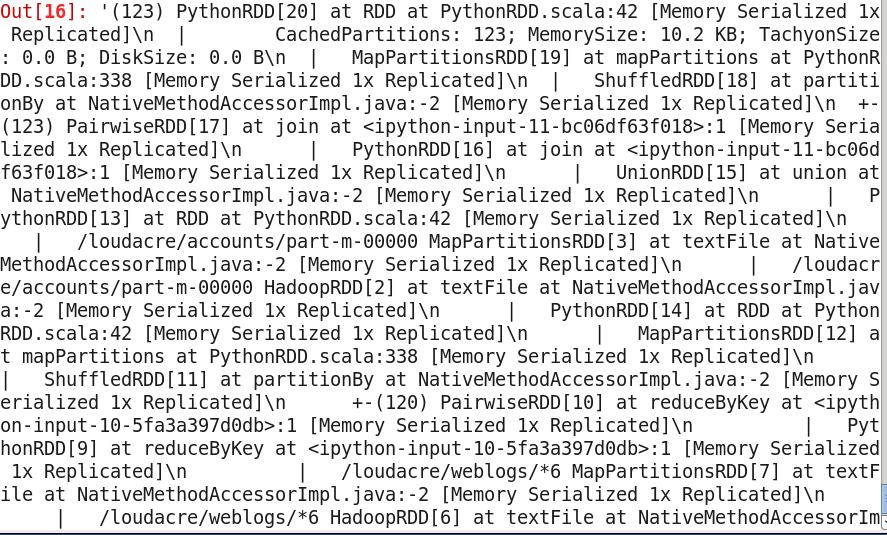
1. No persistence at this stage.



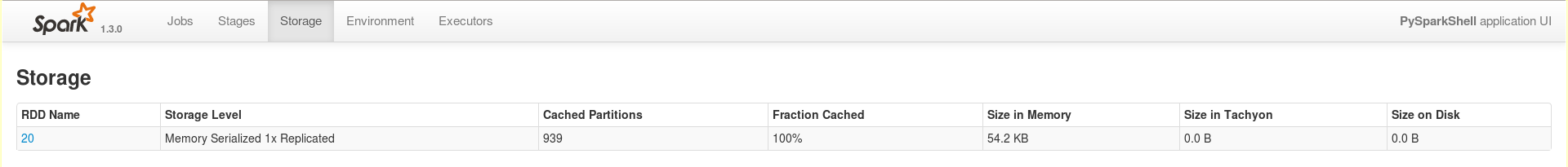
1. Executing count again.



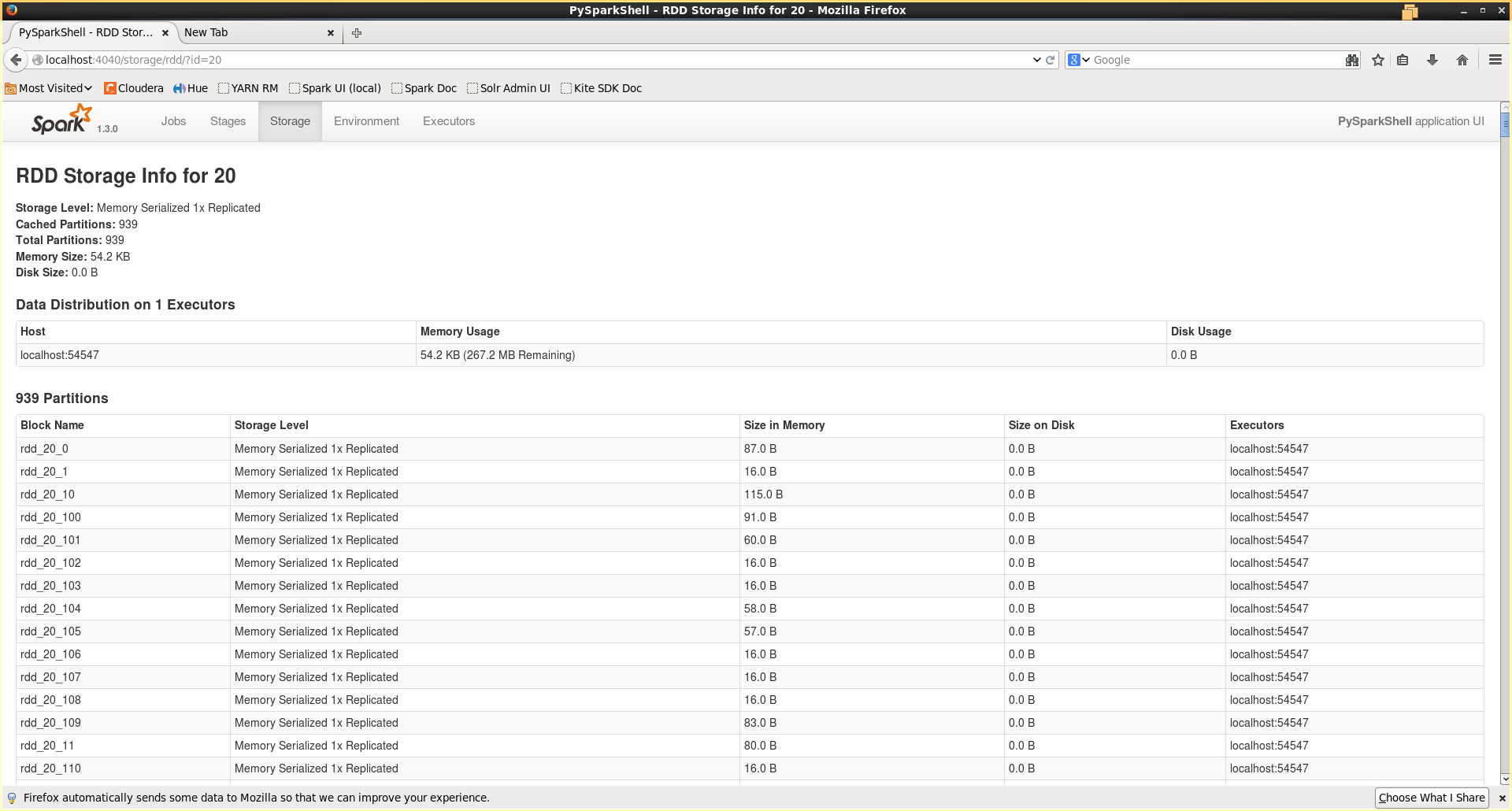
1. Viewing RDD’s to DebugString.



1. RDD persist.

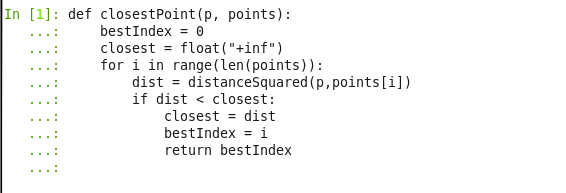


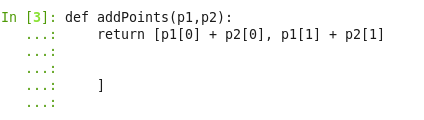
1. Executors tab details.

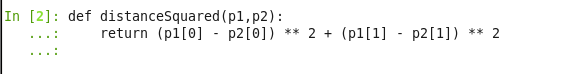


**Chapter 16 Spark Algorithm**

1. Kmeanscoords code.







1. Setting variable K to 5

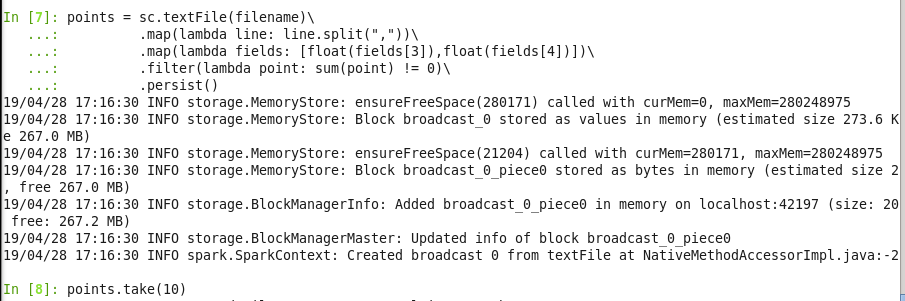


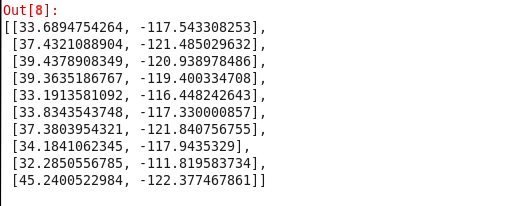


1. Setting converge distance.



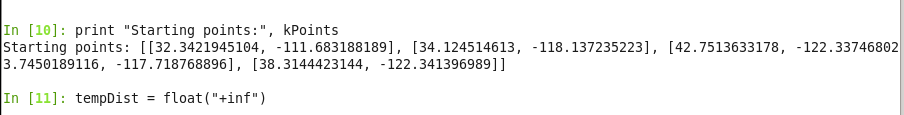
1. Parsing input into long and lat.



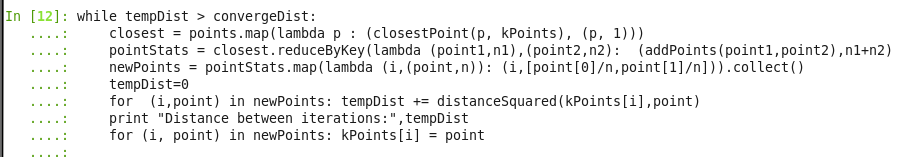


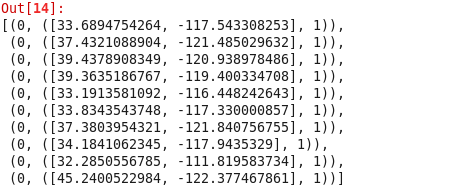
1. Creating k length array called k points by taking random sample k.

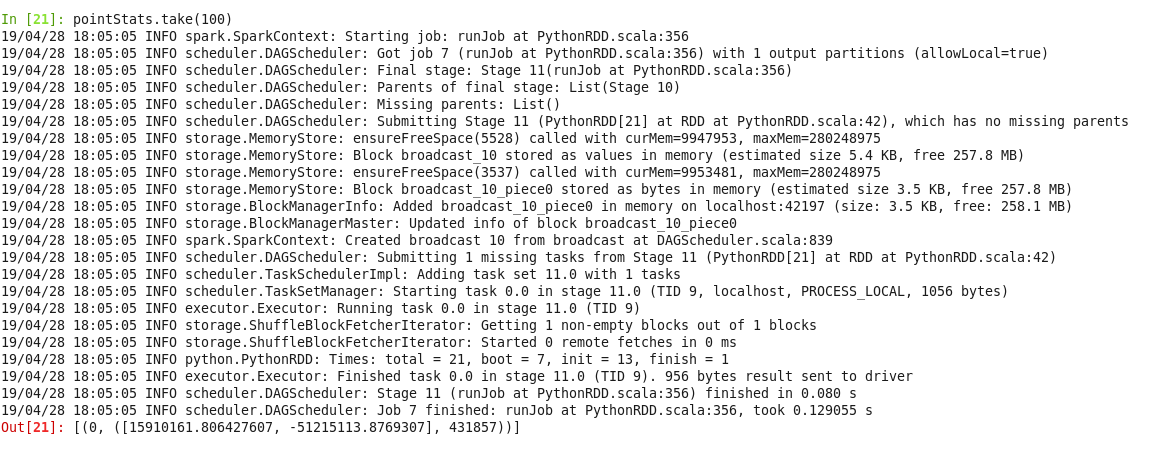


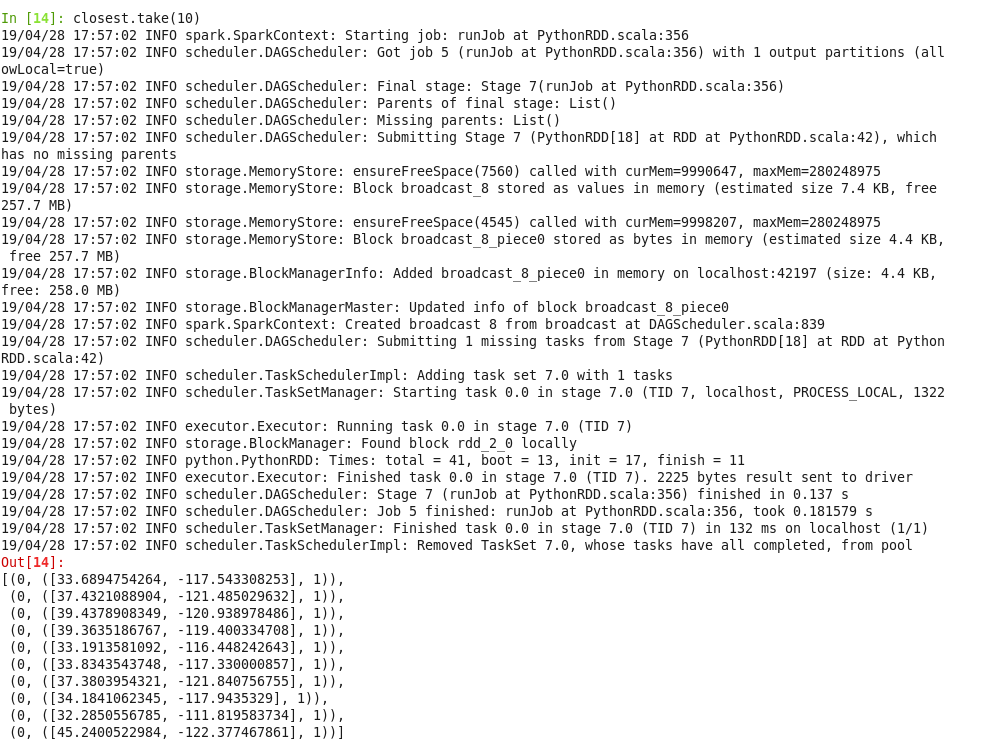


1. Calculating new set of kmeans until the total distance between the means and last is smaller than convergeDist.











1. Displaying final k points.

