CLOUD TECHNOLOGIES – ASSIGNMENT 01

1. STUDENT DETAILS:

a) STUDENT NAME : Karthikeyan Pugazhandhi

b) STUDENT ID : 22267182

c) STUDENT MAIL : karthikeyan.pugazhandhi2@mail.dcu.ie

2. GIT REPOSITORY LINK:

https://github.com/KarthikeyanPugazhandhi01/CloudAssingnment.git

3. DATASET ACQUIRED:

The dataset was obtained via the following link on kaggle.com:

https://www.kaggle.com/datasets/marawanxmamdouh/email-thread-summary-dataset

4. STEPS TAKEN FOR EACH TASK:

a.Installation Of Hadoop, Hive, Pig and Spark:

- By following AWS academic guid and video posted by Michael Scriney I could able to create cluster in AWS.
- My cluster contains application bundle Hadoop, Hive, Pig and Spak.
- I have created 2 core , 1 task using m4large primary, core and task.
- Meantime created S3 bucket to store my input data source.
- Modified security firewall group to include my IP address for SSH script.
- After creating the cluster the cloud9 environment has been setup.
- Now created the EMR instance by including PEM file and writing the below code.

```
voclabs:~/environment $ chmod 600 labsuser.pem
voclabs:~/environment $ ssh -i 'labsuser.pem' hadoop@ec2-23-23-34-198.compute-1.amazonaws.com
The authenticity of host 'ec2-23-23-34-198.compute-1.amazonaws.com (172.31.31.161)' can't be established.
ECDSA key fingerprint is SHA256:Otd7Qq7kdyA3RHWM7JOMWlmLycVIQxlT+BeG33eYLmg.
ECDSA key fingerprint is MD5:1c:5a:99:6e:de:8b:83:76:4c:5d:b5:42:28:4b:1b:65.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-23-23-34-198.compute-1.amazonaws.com,172.31.31.161' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
36 package(s) needed for security, out of 55 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M:::::M:::M   M:::M:::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR:::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE:::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-31-161 ~]$
```

```
[hadoop@ip-172-31-31-161 ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 5b2b8c0c-1a31-47ea-b757-c3bd43816ad6

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
hive>
```

b. Data Extraction:

- The script starts by logging in with secure shell access to Hive on the designated server.
- The email data that was retrieved from an external site (s3://myaqsbucketcloud/) is then stored in an external Hive table called my_Email_data.
- Columns like ID, Subject, EmailDate, EmailFrom, EmailTo, and Body make up this table. The information is kept in a text file in a CSV format.
- The script adds a verification phase that counts the total number of records in the my_Email_data table to confirm the data has been loaded correctly.

```
    > -- Verifying the data
    > SELECT 'Total_Number_of_Raw_Data - ',COUNT(*) FROM my_Email_data;
Query ID = hadoop_20231114140412_78c20fda-d932-4269-adf8-1a29721811f9
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1699969607354_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 25.44 s
--------------------------------------------------------------------------------
OK
Total_Number_of_Raw_Data -      986088
Time taken: 25.834 seconds, Fetched: 1 row(s)
hive>
```

As shown in the output window we have loaded 986088 records

c. Data Cleansing & Purification:

- The script then concentrates on sanitizing the email data.
- By picking different records from the my_Email_data table, it produces a new Hive table entitled my_Email_data_cleaned.
- This step aids in the removal of duplicate entries.
- To ensure that the data is clean, the script counts the total number of records in the my_Email_data_cleaned table.
- To further clean the data, a temporary table named my_Email_data_cleaned_tmp is constructed, which selects only distinct records with IDs less than 241289.
- We choose this figure since the total number of records is 241289. The script then calculates the total number of records in this temporary table and presents the top ten records in descending order, ordered by ID.
- Next, under the name final_email_data_cleaned, the script creates a final cleaned table.
- This table replaces null values with 'N/A,' trims leading and following spaces, and lowercases the subject and body of the text to standardize it.
- Records lacking values in important areas such as ID, EmailDate, EmailFrom, and EmailTo are filtered out of the data.
- This section's script counts how many records are in the final_email_data_cleaned table and shows the top 10 records in descending order of ID.

```
hive> SELECT 'Total_Number_of_Cleaned_Data - ',COUNT(*) FROM final_email_data_cleaned;
OK
Total_Number_of_Cleaned_Data -  20997
Time taken: 0.115 seconds, Fetched: 1 row(s)
hive>
    > SELECT * FROM final_email_data_cleaned
    > ORDER BY ID DESC
    > LIMIT 10;
Query ID = hadoop_20231114141257_32228757-3391-44d0-8d09-982ebd525c03
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1699969607354_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.60 s
--------------------------------------------------------------------------------
OK
999     csfb legal docs 01/08/2000 04:13        sheila glover   ['sara shackleton', 'samantha boyd']    n/a
999     csfb legal docs 04/08/2000 07:44        sara shackleton ['jpeters@andrews-kurth.com']   n/a
999     csfb legal docs 01/09/2000 04:44        sara shackleton ['laurel adams']        n/a
999     csfb legal docs 01/08/2000 02:47        sheila glover   ['sara shackleton', 'samantha boyd']    n/a
998     how's it looking?       08/06/2000 12:00        robin rodrigue  ['ryan watt']   n/a
998     how's it looking?       08/06/2000 12:12        robin rodrigue  ['ryan watt']   n/a
998     how's it looking?       08/06/2000 11:03        ryan watt       ['robin rodrigue']      n/a
998     how's it looking?       08/06/2000 12:18        robin rodrigue  ['ryan watt']   n/a
997     hourly  05/12/2000 07:39        teresa mandola  ['joe stepenovitch']    n/a
997     hourly  05/12/2000 06:33        teresa mandola  ['joe stepenovitch']    n/a
Time taken: 6.932 seconds, Fetched: 10 row(s)
hive>
```

After Cleaning we had around 20997 records and we displayed the top 10 records for sample.

d. Deduction of Ham and Spam

- The script creates a Hive table called spam_words to hold a list of keywords connected to spam emails before dividing emails into ham and spam categories.
- These keywords function as markers to distinguish spam.
- The script's last section focuses on categorizing emails as spam or ham.
- It creates a table called classified_emails, inserts records by determining if the email subject or body contains any of the keywords from the spam_words table.
- Next, the categorization is determined by whether these keywords are present or not.
- In addition to the total number of classified emails, the script offers queries to obtain information about the number of ham and spam emails as well as the top 10 spam and ham accounts according to frequency of occurrence.
- In conclusion, this script uses Hive to handle email classification, data extraction, and cleaning, giving important information about the makeup of the email dataset.

Below query insert the record into classified email table

```
hive>
    > -- Insert classified emails into the classified_emails table
    > INSERT INTO TABLE classified_emails
    > SELECT e.ID, e.EmailFrom, e.Subject, e.Body,
    >        CASE
    >            WHEN (
    >                SELECT COUNT(DISTINCT s.Word)
    >                FROM spam_words s
    >                WHERE e.Subject LIKE CONCAT('%', s.Word, '%')
    >                   OR e.Body LIKE CONCAT('%', s.Word, '%')
    >            ) >= 1 THEN 'spam'
    >            ELSE 'ham'
    >        END AS Classification
    > FROM final_email_data_cleaned e
    > WHERE e.ID IS NOT NULL;
No Stats for default@final_email_data_cleaned, Columns: subject, id, body, emailfrom
Warning: Map Join MAPJOIN[53][bigTable=?] in task 'Reducer 2' is a cross product
Query ID = hadoop_20231116175028_10e1c7e3-2313-40de-be47-618d413aaca7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1700156684801_0001)
```

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      4          4        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 4.18 s
--------------------------------------------------------------------------------
Loading data to table default.classified_emails
OK
Time taken: 6.464 seconds
```

Verifying the result

```
hive>
    > SELECT
    >     COUNT(*) AS total_emails,
    >     COUNT(CASE WHEN Classification = 'ham' THEN 1 END) AS ham_count,
    >     COUNT(CASE WHEN Classification = 'spam' THEN 1 END) AS spam_count
    > FROM
    >     classified_emails;
Query ID = hadoop_20231116175034_4d9c85bb-79c2-4a7d-8f35-dc0804ffe3a3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1700156684801_0001)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 0.51 s
--------------------------------------------------------------------------------
OK
total_emails    ham_count       spam_count
20997   20132   865
Time taken: 1.093 seconds, Fetched: 1 row(s)
hive>
```

As you see we have around 20132 Ham Email and 865 Spam email out of 20997 email.

```
hive> -- Top 10 spam accounts
hive> SELECT EmailFrom, COUNT(*) AS SpamCount
    > FROM classified_emails
    > WHERE Classification = 'spam'
    > GROUP BY EmailFrom
    > ORDER BY SpamCount DESC
    > LIMIT 10;
Query ID = hadoop_20231114143302_e481b640-68ae-43f7-a8af-8b451888f95c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1699969607354_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 7.67 s
--------------------------------------------------------------------------------
OK
emailfrom       spamcount
tana jones      45
sara shackleton 44
kay mann        32
jeff dasovich   30
mark taylor     23
veronica gonzalez       22
shackleton, sara sshackl        15
sally beck      13
matthew lenhart 13
john arnold     13
Time taken: 8.126 seconds, Fetched: 10 row(s)
```

The output shows the top 10 spam account.

```
hive> -- Top 10 ham accounts
hive> SELECT EmailFrom, COUNT(*) AS HamCount
    > FROM classified_emails
    > WHERE Classification = 'ham'
    > GROUP BY EmailFrom
    > ORDER BY HamCount DESC
    > LIMIT 10;
Query ID = hadoop_20231114143654_9197d649-b8bf-495b-a5cd-bedffa96a2fa
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1699969607354_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 7.76 s
--------------------------------------------------------------------------------
OK
emailfrom       hamcount
kay mann        773
tana jones      563
vince j kaminski        555
sara shackleton 552
jeff dasovich   522
chris germany   359
matthew lenhart 317
eric bass       273
carol st clair  267
debra perlingiere       237
Time taken: 8.133 seconds, Fetched: 10 row(s)
hive>
```

The output shows the top 10 Ham account.

e. TFIDF

Since we already found the top 10 spam and ham account using hive, we are using the same table for our advantage and calculating TFIDF using Hive.

- Tokenizing the email's body and subject allows you to count the instances of specific words in the text.
- Divide the combined subject and body of each email into separate words, lowercase them, and count the occurrences.
- To store the results, create a table called tokenization_output and add columns for EmailFrom, term (single words), and term_count (count of each term).

```
hive> CREATE TABLE tokenization_output AS
    > SELECT
    >     EmailFrom,
    >     term,
    >     COUNT(*) AS term_count
    > FROM (
    >     SELECT
    >         EmailFrom,
    >         term
    >     FROM final_email_data_cleaned
    >     LATERAL VIEW explode(split(lower(concat_ws(' ', COALESCE(Subject, ''), COALESCE(Body, ''))), '\\s+')) t AS term
    >     WHERE LENGTH(term) > 0  -- Remove empty terms
    > ) t
    > GROUP BY
    >     EmailFrom, term;
Query ID = hadoop_20231116180846_4e85053a-f85e-4e99-baeb-6a1e43791d93
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1700156684801_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.58 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-13-121.ec2.internal:8020/user/hive/warehouse/tokenization_output
OK
emailfrom       term    term_count
Time taken: 20.06 seconds
```

- Determine each term's Term Frequency (TF) in the tokenization_output table.

- TF is the proportion of a term's count to the email's total number of terms.
- To keep track of the TF values for every term in every email, create a table called tf_output.

```
hive> CREATE TABLE idf_output AS
    > SELECT
    >     term,
    >     LOG(COUNT(DISTINCT EmailFrom) / COUNT(DISTINCT CASE WHEN term_count > 0 THEN EmailFrom END)) AS idf
    > FROM
    >     tokenization_output
    > GROUP BY
    >     term;
Query ID = hadoop_20231116181933_a0f0fc61-5fb2-4ec6-846f-10c5220442fe
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1700156684801_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 1.46 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-13-121.ec2.internal:8020/user/hive/warehouse/idf_output
OK
term    idf
Time taken: 2.287 seconds
```

- For every term, determine the Inverse Document Frequen Based on the quantity of spam emails, determine the top ten spam accounts.
- To keep track of the most popular spam accounts and the number of spam emails associated with each account, create a table called top_spam_accounts.

```
hive> CREATE TABLE idf_output AS
    > SELECT
    >     term,
    >     LOG(COUNT(DISTINCT EmailFrom) / COUNT(DISTINCT CASE WHEN term_count > 0 THEN EmailFrom END)) AS idf
    > FROM
    >     tokenization_output
    > GROUP BY
    >     term;
Query ID = hadoop_20231116182231_8fea65f9-1a2d-4cac-9784-d5f35b0746ad
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1700156684801_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 8.55 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-13-121.ec2.internal:8020/user/hive/warehouse/idf_output
OK
term    idf
Time taken: 9.299 seconds
```

- Establish top_spam_keywords. Tabular Data
- Determine the top 10 keywords (based on TF-IDF values) for each of the top 10 spam accounts.
- To keep track of the top keywords for each of the top 10 spam accounts, create a table called top_spam_keywords. Similary we for (IDF).
- The logarithm of the ratio of all distinct emails to all emails containing the term is called the indistinguishable degree of fit (IDF).
- Make a table called idf_output to hold each term's IDF values.
- For every term in every email, determine the TF-IDF (Term Frequency-Inverse Document Frequency).

- The IDF value from the idf_output table should be multiplied by the TF value from the tf_output table.
- To keep track of the TF-IDF values for every term in every email, create a table called tfidf output.
- Similary we did for ham account and displayed the result.

Top 10 spam account with top 10 spam keyword

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 3 ..... container     SUCCEEDED      1          1        0        0       0       0
Map 1 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 13.82 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-13-121.ec2.internal:8020/user/hive/warehouse/top_spam_keywords
OK
ts.emailfrom    tsfk.term       tsfk.tfidf
Time taken: 15.04 seconds
hive> SET hive.cli.print.header=true;
hive> -- Display the results
hive> SELECT * FROM top_spam_keywords;
OK
top_spam_keywords.emailfrom     top_spam_keywords.term  top_spam_keywords.tfidf
jeff dasovich   brazil  0.0
jeff dasovich   briefing        0.0
jeff dasovich   broadband       0.0
jeff dasovich   buddy.  0.0
jeff dasovich   bundle  0.0
jeff dasovich   bush    0.0
jeff dasovich   by      0.0
jeff dasovich   ca      0.0
jeff dasovich   california/west 0.0
jeff dasovich   can     0.0
sally beck      internal        0.0
sally beck      initiatives.    0.0
sally beck      informed        0.0
sally beck      information     0.0
sally beck      influence       0.0
sally beck      in      0.0
sally beck      impact  0.0
sally beck      hiring  0.0
sally beck      handling        0.0
```

```
sally beck      group   0.0
sara shackleton panus   0.0
sara shackleton paso    0.0
sara shackleton pc      0.0
sara shackleton pec     0.0
sara shackleton perez   0.0
sara shackleton performance     0.0
sara shackleton platforms       0.0
sara shackleton plc's   0.0
sara shackleton please  0.0
sara shackleton prepay  0.0
tana jones      try     0.0
tana jones      trust   0.0
tana jones      treaties        0.0
tana jones      transalta       0.0
tana jones      tradespark      0.0
tana jones      trade   0.0
tana jones      too     0.0
tana jones      to      0.0
tana jones      thursday,       0.0
tana jones      thursday        0.0
john arnold     unsubscribe     0.0
john arnold     trading 0.0
john arnold     to      0.0
john arnold     tickets 0.0
john arnold     talk    0.0
john arnold     swaps   0.0
john arnold     survey/information      0.0
john arnold     super   0.0
john arnold     started 0.0
john arnold     sensitive:      0.0
kay mann        lake    0.0
kay mann        kay     0.0
kay mann        june    0.0
kay mann        job     0.0
kay mann        items   0.0
kay mann        it      0.0
```

Top 10 Ham account with 10 ham keywords

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 14.52 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-13-121.ec2.internal:8020/user/hive/warehouse/top_ham_keywords
OK
th.emailfrom    thfk.term       thfk.tfidf
Time taken: 15.577 seconds
hive>
    >
    > SET hive.cli.print.header=true;
hive> -- Display the results
hive> SELECT * FROM top_ham_keywords;
OK
top_ham_keywords.emailfrom      top_ham_keywords.term   top_ham_keywords.tfidf
debra perlingiere       jm      0.0
debra perlingiere       info    0.0
debra perlingiere       inc.    0.0
debra perlingiere       hey!    0.0
debra perlingiere       help!   0.0
debra perlingiere       help    0.0
debra perlingiere       hello's 0.0
debra perlingiere       happy   0.0
debra perlingiere       gulf    0.0
debra perlingiere       glendale        0.0
jeff dasovich   worry   0.0
jeff dasovich   talking 0.0
jeff dasovich   "e")    0.0
jeff dasovich   #1      0.0
jeff dasovich   (and    0.0
jeff dasovich   --      0.0
```

```
jeff dasovich   -sources        0.0
jeff dasovich   10      0.0
jeff dasovich   12-     0.0
jeff dasovich   13      0.0
sara shackleton deutsche        0.0
sara shackleton duke    0.0
sara shackleton duty    0.0
sara shackleton e-mail  0.0
sara shackleton ect     0.0
sara shackleton edison  0.0
sara shackleton ees     0.0
sara shackleton electric        0.0
sara shackleton email_verification      0.0
sara shackleton ena     0.0
tana jones      important-please        0.0
tana jones      in      0.0
tana jones      inc.    0.0
tana jones      inc.,   0.0
tana jones      income  0.0
tana jones      information     0.0
tana jones      invoice 0.0
tana jones      it      0.0
tana jones      j.      0.0
tana jones      japan   0.0
vince j kaminski        tournament      0.0
vince j kaminski        tony    0.0
vince j kaminski        to_do   0.0
vince j kaminski        to      0.0
vince j kaminski        time-series     0.0
vince j kaminski        these   0.0
vince j kaminski        team    0.0
vince j kaminski        tanya's 0.0
vince j kaminski        tails"  0.0
vince j kaminski        tage    0.0
carol st clair  holiday 0.0
carol st clair  important       0.0
```