Assignment 2 Report

Steps Followed:

First Downloaded both the log files extracted them and copied it to a directory from where I started my jupyter notebook.

Installed pyspark “pip install pyspark” , then setup java version "1.8.0_441".

1. Started a ipynb file started spark session

```
import pyspark

from pyspark.sql import SparkSession

from pyspark import SparkContext

from pyspark import SparkConf
conf = pyspark.SparkConf() \
.setMaster("local[*]") \
.set("spark.executor.heartbeatInterval", "60s") \

 .set("spark.network.timeout", "300s") \
.set("spark.sql.shuffle.partitions", "200") \
.set("spark.executor.memory", "15G") \
.set("spark.driver.memory", "15G") \
.set("spark.driver.maxResultSize", "10G")
# Initialize SparkSession

spark = SparkSession.builder \
.appName("BDS-Assignment-2") \
 .config(conf=conf) \
.getOrCreate()
```

2. Loading log files and merging them using union operation

```
aug_file = spark.read.text("access_log_Aug95")
jul_file = spark.read.text("access_log_Jul95")
combined_logs = aug_file.union(jul_file)
```

3. Parsing the combined log file since its a log file need to derive data into dataset table so using regular expression from pyspark. Issue faced was that we did get any structured data ,since log files are unstructured had to use regex to break down and separate into dataframe table.

from pyspark.sql.functions import regexp_extract, col, to_timestamp, when,count, countDistinct, date_format, to_date, desc ,avg

log_pattern = r'^(\S+) (\S+) (\S+) [([\w:/]+\s[+-]\d{4})] "(\S+) (\S+) (\S+)" (\d{3}) (\S+)$'

parsed_logs = combined_logs.select(regexp_extract(col("value"), log_pattern, 1).alias("remotehost"), regexp_extract(col("value"), log_pattern, 2).alias("rfc931"), regexp_extract(col("value"), log_pattern, 3).alias("authuser"), regexp_extract(col("value"), log_pattern, 4).alias("date"), regexp_extract(col("value"), log_pattern, 5).alias("method"), regexp_extract(col("value"), log_pattern, 6).alias("endpoint"), regexp_extract(col("value"), log_pattern, 7).alias("protocol"), regexp_extract(col("value"), log_pattern, 8).alias("status"), regexp_extract(col("value"), log_pattern, 9).alias("bytes"))

parsed_logs.show(truncate=False)

```
+--------------------------+------+--------+--------------------------+------+----------
----------------------------------------+--------+------+-----+
|remotehost                |rfc931|authuser|date                      |method|endpoint
|protocol|status|bytes|
+--------------------------+------+--------+--------------------------+------+----------
----------------------------------------+--------+------+-----+
|in24.inetnebr.com         |-     |-       |01/Aug/1995:00:00:01 -0400|GET
|/shuttle/missions/sts-68/news/sts-68-mcc-05.txt    |HTTP/1.0|200   |1839 |
|uplherc.upl.com           |-     |-       |01/Aug/1995:00:00:07 -0400|GET   |/
|HTTP/1.0|304   |0    |
|uplherc.upl.com           |-     |-       |01/Aug/1995:00:00:08 -0400|GET
|/images/ksclogo-medium.gif                        |HTTP/1.0|304   |0    |
```

3. filling null in rfc , authuser and bytes column where ever its empty where ever the value is “-” replaceing with null

parsed_logs = parsed_logs.withColumn("rfc931", when(col("rfc931") != "-", col("rfc931")).otherwise(None))

.withColumn("authuser", when(col("authuser") != "-", col("authuser")).otherwise(None))

.withColumn("bytes", when(col("bytes") != "-", col("bytes")).otherwise(None))

Removing empty rows using na.drop **cleaned_logs = parsed_logs.na.drop(how="all")**

converting status to int, bytes to int and date to proper date format
cleaned_logs = cleaned_logs.withColumn("status", col("status").cast("int")) .withColumn("bytes", col("bytes").cast("int")) .withColumn("date", to_timestamp(col("date"), "dd/MMM/yyyy:HH:mm:ss Z"))

cleaned_logs.show()

```
+--------------------+------+--------+-------------------+------+--------------------+---
-----+------+-----+
|          remotehost|rfc931|authuser|               date|method|
endpoint|protocol|status|bytes|
+--------------------+------+--------+-------------------+------+--------------------+---
-----+------+-----+
|   in24.inetnebr.com|  NULL|    NULL|1995-08-01 09:30:01|
GET|/shuttle/missions...|HTTP/1.0|   200| 1839|
|     uplherc.upl.com|  NULL|    NULL|1995-08-01 09:30:07|   GET|
/|HTTP/1.0|   304|    0|
```

4. Count of total log records
total_log_records = cleaned_logs.count()
print(f"Total log records: {total_log_records}")

```
Total log records: 3461613
```

5. Count of unique hosts **unique_hosts_count = cleaned_logs.select(countDistinct("remotehost")).collect()[0][0] print(f"Number of unique hosts: {unique_hosts_count}")**

```
Number of unique hosts: 137827
```

6. Date wise unique host counts
datewise_unique_hosts = cleaned_logs.groupBy(to_date("date").alias("date")) .agg(countDistinct("remotehost") .alias("unique_hosts")) .orderBy("date") .withColumn("date", date_format(col("date"), "dd-MMM-yyyy")) print("Date-wise unique host counts:")
datewise_unique_hosts.show(truncate=False)

```
Date-wise unique host counts:
+-----------+------------+
|date       |unique_hosts|
+-----------+------------+
|NULL       |1           |
|01-Jul-1995|2854        |
|02-Jul-1995|4887        |
|03-Jul-1995|6535        |
|04-Jul-1995|6514        |
|05-Jul-1995|6426        |
|06-Jul-1995|7714        |
|07-Jul-1995|7639        |
```

```
|08-Jul-1995|4053        |
|09-Jul-1995|2608        |
|10-Jul-1995|3860        |
|11-Jul-1995|4631        |
|12-Jul-1995|5212        |
|13-Jul-1995|6780        |
|14-Jul-1995|5797        |
|15-Jul-1995|4013        |
|16-Jul-1995|2885        |
|17-Jul-1995|4381        |
|18-Jul-1995|4673        |
|19-Jul-1995|4968        |
+-----------+------------+
only showing top 20 rows
```

7. Average Requests per Host per Day

requests_per_host_per_day = cleaned_logs.groupBy(to_date("date").alias("date"), "remotehost") .agg(count("*").alias("requests")) .groupBy("date") .agg(avg("requests"). alias("avg_requests_per_host")) .orderBy("date") .withColumn("date", date_format(col("date"), "dd-MMM-yyyy"))

print("Average requests per host per day:")

requests_per_host_per_day.show(truncate=False)

```
Average requests per host per day:
+-----------+--------------------+
|date       |avg_requests_per_host|
+-----------+--------------------+
|NULL       |8314.0              |
|01-Jul-1995|11.978626489138051  |
|02-Jul-1995|12.23449969306323   |
|03-Jul-1995|12.312777352716143  |
|04-Jul-1995|12.137549892539147  |
|05-Jul-1995|12.724867724867725  |
|06-Jul-1995|12.629504796473944  |
|07-Jul-1995|13.031286817646288  |
|08-Jul-1995|13.374537379718728  |
|09-Jul-1995|13.70935582822086   |
|10-Jul-1995|14.543264248704663  |
|11-Jul-1995|16.636579572446557  |
|12-Jul-1995|16.96412125863392   |
|13-Jul-1995|19.796755162241887  |
|14-Jul-1995|15.630671036743143  |
|15-Jul-1995|14.399700971841515  |
|16-Jul-1995|15.2315424610052    |
|17-Jul-1995|15.005021684546907  |
|18-Jul-1995|14.612454526000429  |
|19-Jul-1995|14.457528180354267  |
+-----------+--------------------+
only showing top 20 rows
```

8. Number of 404 Response Codes **count_404 = cleaned_logs.filter(col("status") == 404).count()**

print(f"Number of 404 responses: {count_404}") `Number of 404 responses: 20621`

9. Top 15 Endpoints with 404 Responses

```
top_404_endpoints = cleaned_logs.filter(col("status") ==
404) .groupBy("endpoint") .agg(count("*").alias("404_count")) .orderBy(desc("404_cou
nt")) .limit(15)
print("Top 15 endpoints with 404 responses:")
top_404_endpoints.show(truncate=False)
```

```
Top 15 endpoints with 404 responses:
+-----------------------------------------------------------------+---------+
|endpoint                                                         |404_count|
+-----------------------------------------------------------------+---------+
|/pub/winvn/readme.txt                                            |2004     |
|/pub/winvn/release.txt                                           |1732     |
|/shuttle/missions/STS-69/mission-STS-69.html                     |682      |
|/shuttle/missions/sts-68/ksc-upclose.gif                         |426      |
|/history/apollo/a-001/a-001-patch-small.gif                      |384      |
|/history/apollo/sa-1/sa-1-patch-small.gif                        |383      |
|/://spacelink.msfc.nasa.gov                                      |381      |
|/images/crawlerway-logo.gif                                      |374      |
|/elv/DELTA/uncons.htm                                            |372      |
|/history/apollo/pad-abort-test-1/pad-abort-test-1-patch-small.gif|359      |
|/images/nasa-logo.gif                                            |319      |
|/shuttle/resources/orbiters/atlantis.gif                         |310      |
|/history/apollo/apollo-13.html                                   |304      |
|/shuttle/resources/orbiters/discovery.gif                        |262      |
|/shuttle/missions/sts-71/images/KSC-95EC-0916.txt                |190      |
+-----------------------------------------------------------------+---------+
```

10. Top 15 Hosts with 404 Responses

```
top_404_hosts = parsed_logs.filter(col("status") ==
404) .groupBy("remotehost") .agg(count("*").alias("404_count")) .orderBy(desc("404_c
ount")) .limit(15)
print("Top 15 hosts with 404 responses:")
top_404_hosts.show(truncate=False)
```

```
Top 15 hosts with 404 responses:
+--------------------------+---------+
|remotehost                |404_count|
+--------------------------+---------+
|hoohoo.ncsa.uiuc.edu      |251      |
|piweba3y.prodigy.com      |157      |
|jbiagioni.npt.nuwc.navy.mil|132      |
|piweba1y.prodigy.com      |114      |
|www-d4.proxy.aol.com      |91       |
|piweba4y.prodigy.com      |86       |
|scooter.pa-x.dec.com      |69       |
|www-d1.proxy.aol.com      |64       |
|phaelon.ksc.nasa.gov      |64       |
|www-b4.proxy.aol.com      |62       |
|dialip-217.den.mmc.com    |62       |
|www-b3.proxy.aol.com      |61       |
```

```
|www-a2.proxy.aol.com       |60       |
|piweba2y.prodigy.com       |59       |
|www-d2.proxy.aol.com       |59       |
+---------------------------+---------+
```

11. Stop spark **spark.stop()**