Problem Set D Submission Form

Overview

Your Name	Hendi Kushta
Your SU Email	hkushta@syr.edu

Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screen shot guidelines from the first homework.

Remember as you complete the homework it is not only about getting it right / correct. We will discuss the answers in class so it's important to articulate anything you would like to contribute to the discussion in your answer:

- If you feel the question is vague, include any assumptions you've made.
- If you feel the answer requires interpretation or justification provide it.
- If you do not know the answer to the question, articulate what you tried and how you are stuck.
- Highlight any doubts or questions you would like me to review.

This how you receive credit for answering questions which might not be correct. In addition, you must complete the reflection portion of the homework assignment for full credit. Since most answers will be similar this is an important part of your individual submission.

Complete Part II of this document first, then go back and complete the Reflection in Part I.

Part I - Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment you must be as descriptive and personal as possible with your reflection.

1. As you completed this assignment, identify what you learned.

   Manage data in object storage like AWS S3, or Minio. Use Apache Spark to perform extract, transform and load of semi-structured data. Use Apache Spark SQL to perform a basic analysis of the data.t the end of this lab, you should be able to:

2. What barriers or challenges did you encounter while completing this assignment?

3. How prepared were you to complete this assignment? What can you do to be better prepared?

4. Rate your comfort level with this week’s material. Use the rubric provided.

4 ==> I understand this material and can explain it to others.
3 ==> I understand this material.
2 ==> I somewhat understand the material but sometimes need guidance from others.
1 ==> I understand very little of this material and need extra help.

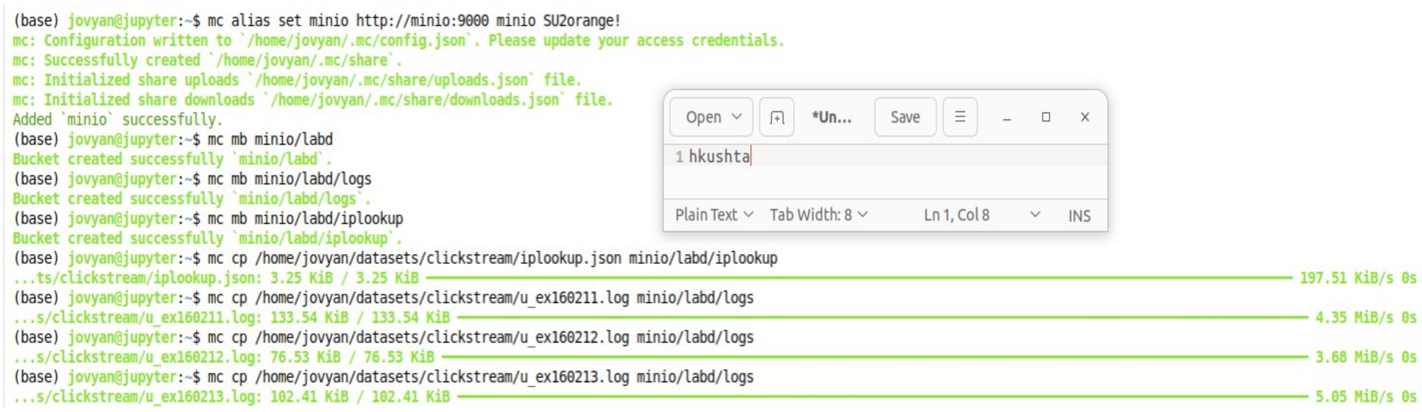
Part II – Questions

Paste your answers to the Exercises found in the lab document. Make sure to include your netid in any screenshots you provide. If the question asks for commands, only include those commands which are necessary to complete the tasks. Number each answer.

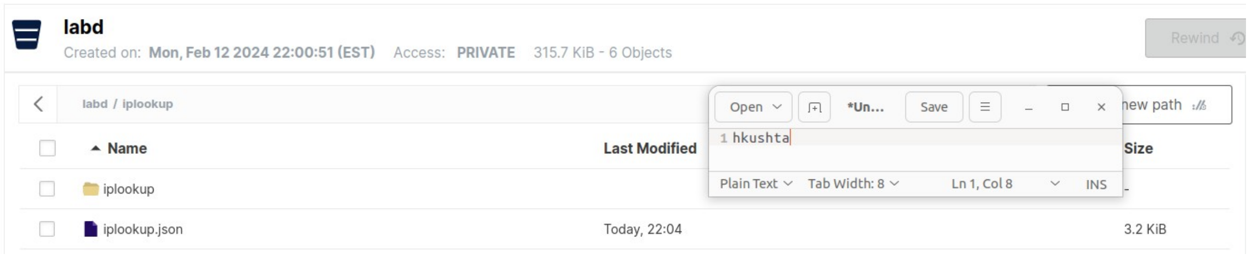
1. Connect to the minio client. Create an alias to your minio server, named **ms**. Create a bucket **labd**. Inside the bucket, create folders **iplookup** and **logs**. Copy the 3 log files from **/datasets/clickstream** to the minio **logs** folder. Copy **iplookup.json** to the **iplookup** folder.

When you are finished you should have the following structure. Provide a list of commands necessary to complete this task. And include screenshots to show the files are there.

```
ms
|
|__ labd
      |
      |__ iplookup
      |   |
      |   |__ iplookup.json
      |
      |__logs
         |
         |__ u_ex160211.log, u_ex160212.log, u_ex160213.log
```



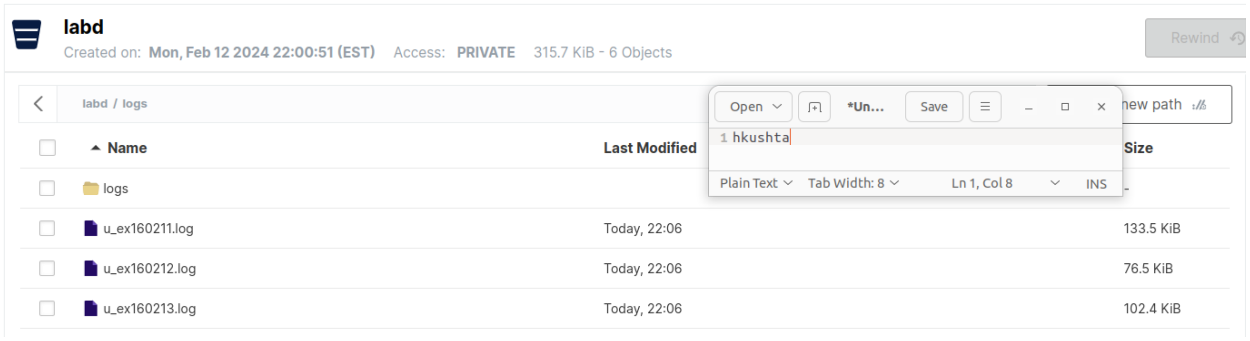
```
(base) jovyan@jupyter:~$ mc alias set minio http://minio:9000 minio SU2orange!
mc: Configuration written to `/home/jovyan/.mc/config.json`. Please update your access credentials.
mc: Successfully created `/home/jovyan/.mc/share`.
mc: Initialized share uploads `/home/jovyan/.mc/share/uploads.json` file.
mc: Initialized share downloads `/home/jovyan/.mc/share/downloads.json` file.
Added `minio` successfully.
(base) jovyan@jupyter:~$ mc mb minio/labd
Bucket created successfully `minio/labd`.
(base) jovyan@jupyter:~$ mc mb minio/labd/logs
Bucket created successfully `minio/labd/logs`.
(base) jovyan@jupyter:~$ mc mb minio/labd/iplookup
Bucket created successfully `minio/labd/iplookup`.
(base) jovyan@jupyter:~$ mc cp /home/jovyan/datasets/clickstream/iplookup.json minio/labd/iplookup
...ts/clickstream/iplookup.json: 3.25 KiB / 3.25 KiB ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 197.51 KiB/s 0s
(base) jovyan@jupyter:~$ mc cp /home/jovyan/datasets/clickstream/u_ex160211.log minio/labd/logs
...s/clickstream/u_ex160211.log: 133.54 KiB / 133.54 KiB ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.35 MiB/s 0s
(base) jovyan@jupyter:~$ mc cp /home/jovyan/datasets/clickstream/u_ex160212.log minio/labd/logs
...s/clickstream/u_ex160212.log: 76.53 KiB / 76.53 KiB ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.68 MiB/s 0s
(base) jovyan@jupyter:~$ mc cp /home/jovyan/datasets/clickstream/u_ex160213.log minio/labd/logs
...s/clickstream/u_ex160213.log: 102.41 KiB / 102.41 KiB ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.05 MiB/s 0s
```


labd
Created on: Mon, Feb 12 2024 22:00:51 (EST) Access: PRIVATE 315.7 KiB - 6 Objects

labd / iplookup

1 hkushta

Name	Last Modified	Size
iplookup		
iplookup.json	Today, 22:04	3.2 KiB

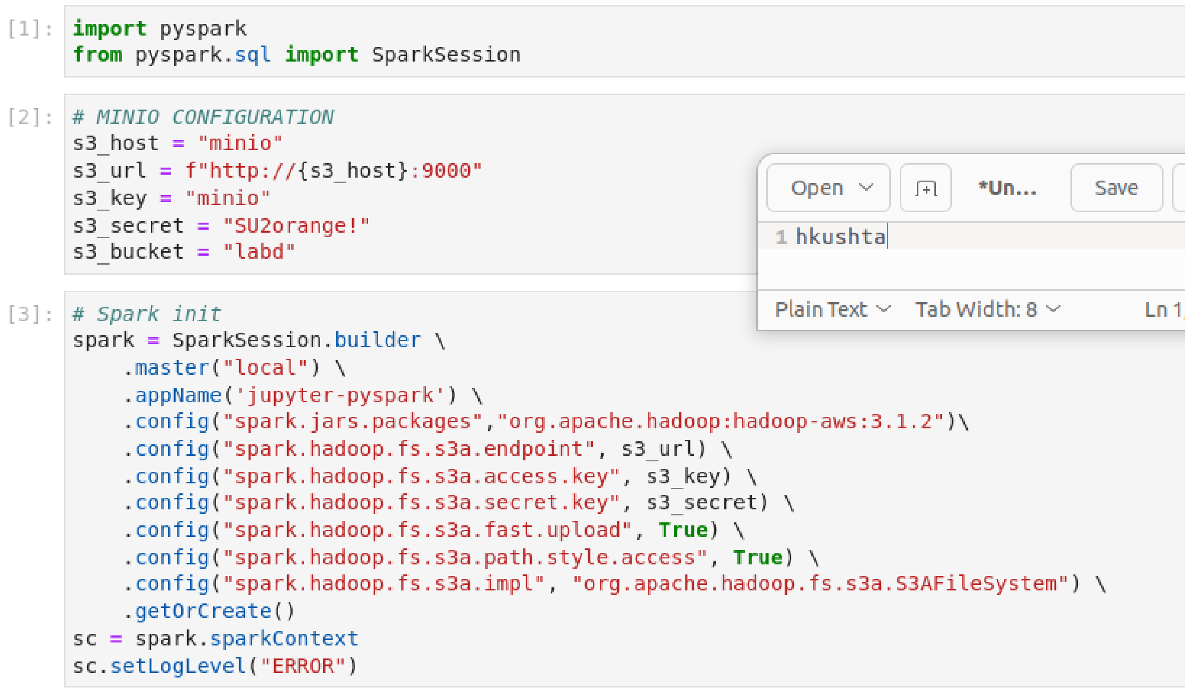

labd
Created on: Mon, Feb 12 2024 22:00:51 (EST) Access: PRIVATE 315.7 KiB - 6 Objects

labd / logs

1 hkushta

Name	Last Modified	Size
logs		
u_ex160211.log	Today, 22:06	133.5 KiB
u_ex160212.log	Today, 22:06	76.5 KiB
u_ex160213.log	Today, 22:06	102.4 KiB

2. Create a new Spark notebook called **labd.ipynb** write (or copy and edit) Spark code to setup the Spark session. Make sure your spark session supports Minio access and include the **hadoop-aws** Spark Jar package. Provide a screenshot of your code and the output.
NOTE: You do not need Hive support.

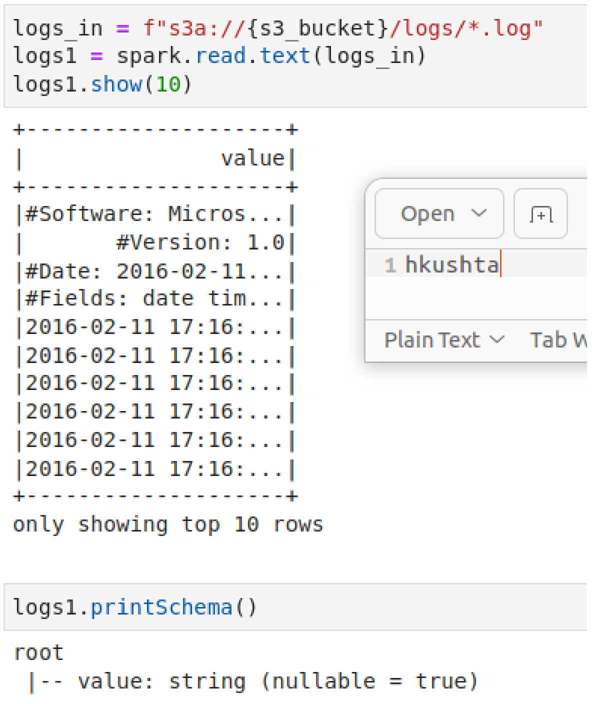

```
[1]: import pyspark
     from pyspark.sql import SparkSession
```

```
[2]: # MINIO CONFIGURATION
     s3_host = "minio"
     s3_url = f"http://{s3_host}:9000"
     s3_key = "minio"
     s3_secret = "SU2orange!"
     s3_bucket = "labd"
```

1 hkushta

```
[3]: # Spark init
     spark = SparkSession.builder \
         .master("local") \
         .appName('jupyter-pyspark') \
         .config("spark.jars.packages","org.apache.hadoop:hadoop-aws:3.1.2")\
         .config("spark.hadoop.fs.s3a.endpoint", s3_url) \
         .config("spark.hadoop.fs.s3a.access.key", s3_key) \
         .config("spark.hadoop.fs.s3a.secret.key", s3_secret) \
         .config("spark.hadoop.fs.s3a.fast.upload", True) \
         .config("spark.hadoop.fs.s3a.path.style.access", True) \
         .config("spark.hadoop.fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem") \
         .getOrCreate()
     sc = spark.sparkContext
     sc.setLogLevel("ERROR")
```

3. Write Spark code to load logs from Minio **labd/logs** into a dataframe **logs1** using **spark.read.text**. print the schema and show 10 rows from the DataFrame. Screenshot the code and output.



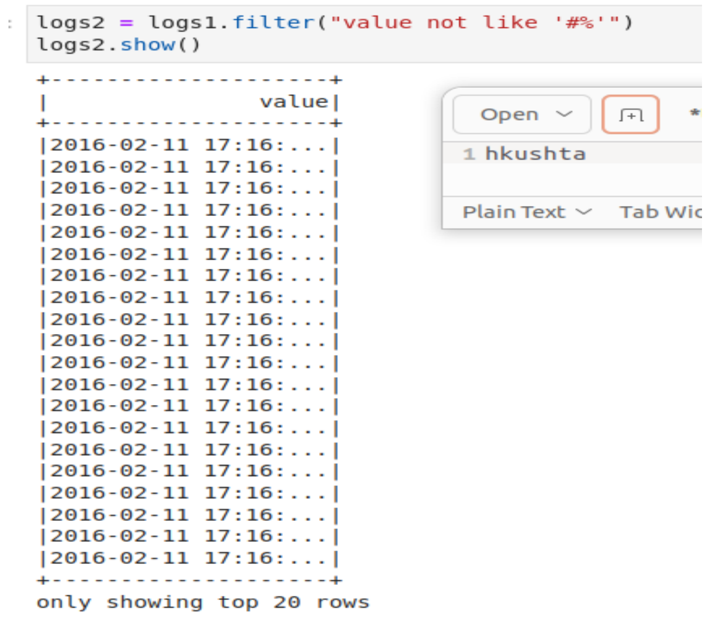
```
logs_in = f"s3a://{s3_bucket}/logs/*.log"
logs1 = spark.read.text(logs_in)
logs1.show(10)
```

```
+--------------------+
|               value|
+--------------------+
|#Software: Micros...|
|       #Version: 1.0|
|#Date: 2016-02-11...|
|#Fields: date tim...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
+--------------------+
only showing top 10 rows
```

```
logs1.printSchema()
```

```
root
 |-- value: string (nullable = true)
```

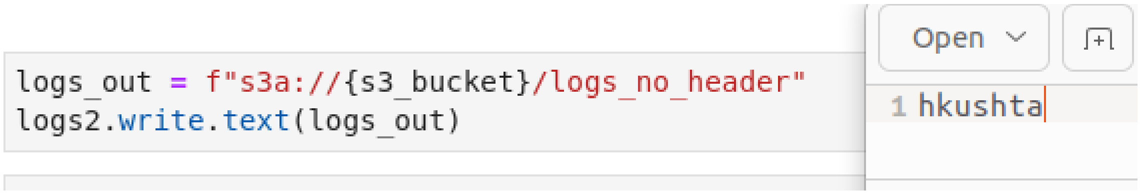
4. We need to remove the rows with **#** in front of them, as these are comments in the web server log files. Use the **filter()** function to do this, and save the results into dataframe **logs2**. Show the code and output in your screenshot.



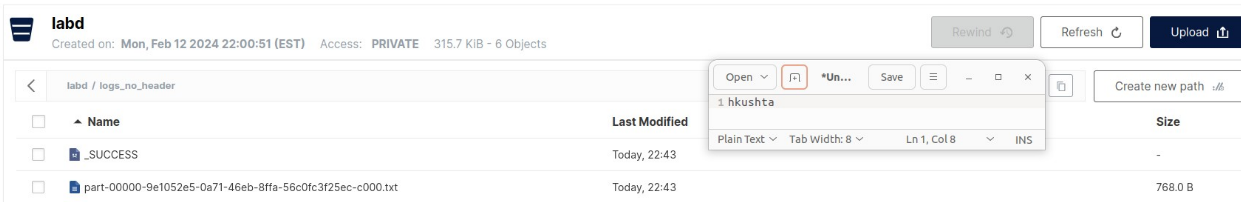
```
logs2 = logs1.filter("value not like '#%'")
logs2.show()
```

```
+--------------------+
|               value|
+--------------------+
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
|2016-02-11 17:16:...|
+--------------------+
only showing top 20 rows
```

5. Write back your **logs2** to Minio. Use the **text** format and call the file **logs-no-header.** Include a screenshot of the code and output.



```
logs_out = f"s3a://{s3_bucket}/logs_no_header"
logs2.write.text(logs_out)
```



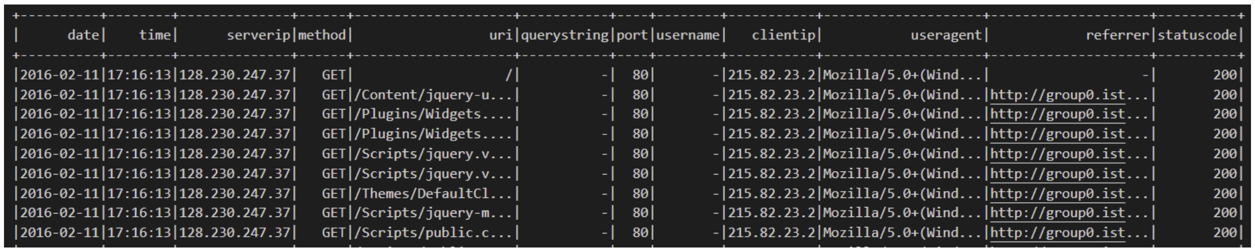
labd
Created on: Mon, Feb 12 2024 22:00:51 (EST)   Access: PRIVATE   315.7 KiB - 6 Objects

labd / logs_no_header

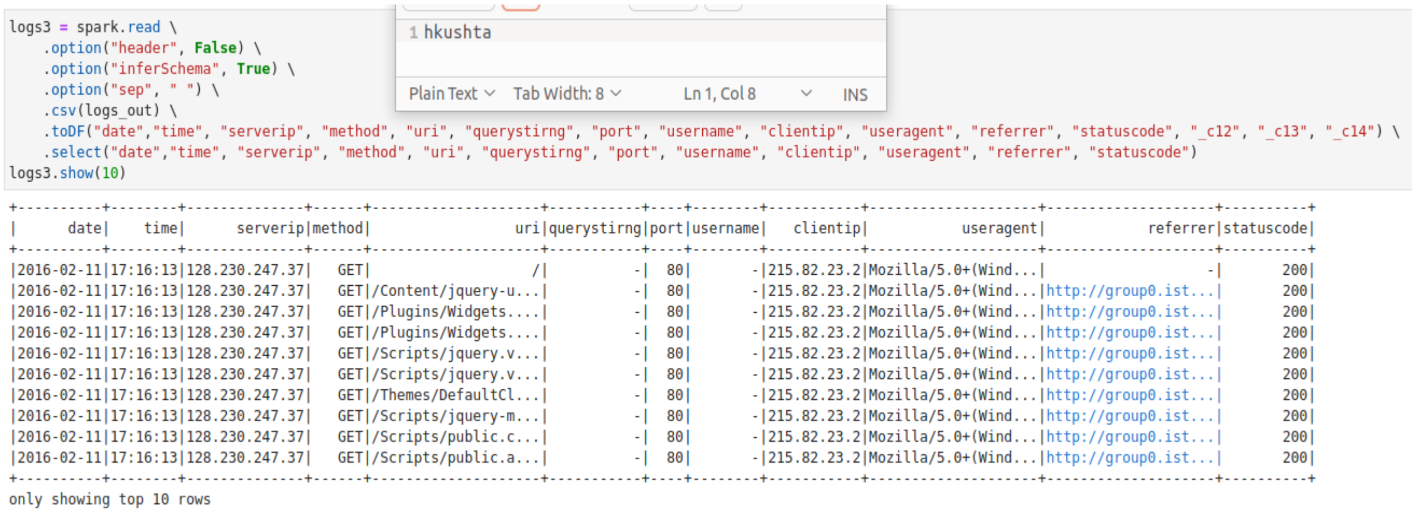
Name	Last Modified	Size
_SUCCESS	Today, 22:43	-
part-00000-9e1052e5-0a71-46eb-8ffa-56c0fc3f25ec-c000.txt	Today, 22:43	768.0 B

6. Read in the **logs-no-header** this time using **csv** to delimit on a space into **logs3**. Add headers (date,time, serverip, method, uri, querystirng, port, username, clientip, useragent, referrer, statuscode), and provide an output of the schema and the first couple of rows from the Dataframe itself in the screenshot.

Here is what the DataFrame Should look like:

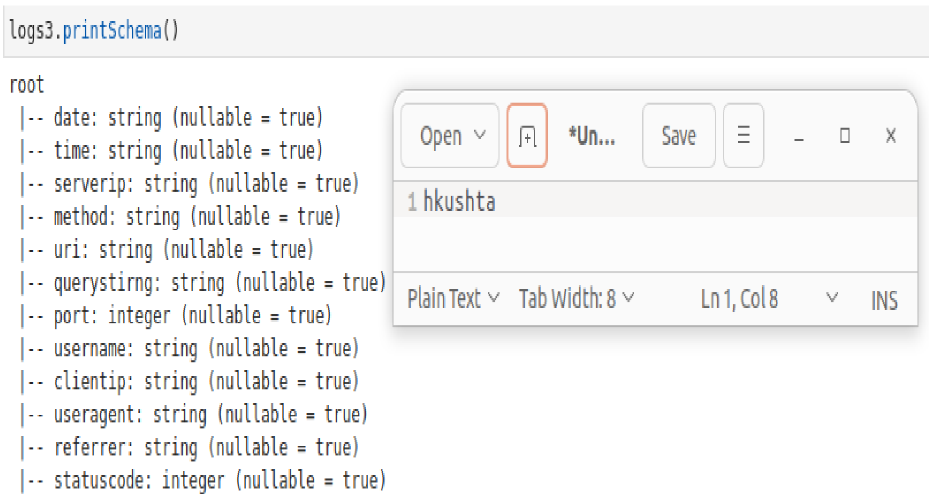


```
+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|      date|    time|      serverip|method|                 uri|querystring|port|username|   clientip|           useragent|            referrer|statuscode|
+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|2016-02-11|17:16:13|128.230.247.37|   GET|                   /|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|                   -|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Content/jquery-u...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery.v...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery.v...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Themes/DefaultCl...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery-m...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/public.c...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
```



```
logs3 = spark.read \
    .option("header", False) \
    .option("inferSchema", True) \
    .option("sep", " ") \
    .csv(logs_out) \
    .toDF("date","time", "serverip", "method", "uri", "querystirng", "port", "username", "clientip", "useragent", "referrer", "statuscode", "_c12", "_c13", "_c14") \
    .select("date","time", "serverip", "method", "uri", "querystirng", "port", "username", "clientip", "useragent", "referrer", "statuscode")
logs3.show(10)
```

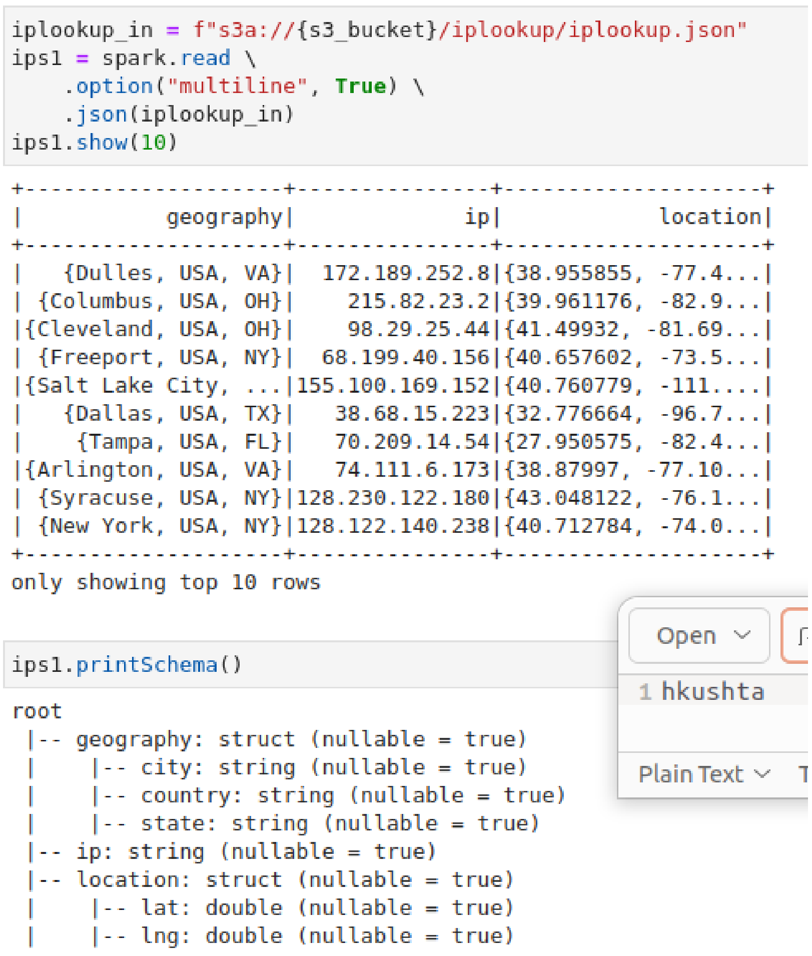
```
+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|      date|    time|      serverip|method|                 uri|querystirng|port|username|   clientip|           useragent|            referrer|statuscode|
+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|2016-02-11|17:16:13|128.230.247.37|   GET|                   /|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|                   -|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Content/jquery-u...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery.v...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery.v...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Themes/DefaultCl...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery-m...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/public.c...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/public.a...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
only showing top 10 rows
```



```
logs3.printSchema()

root
 |-- date: string (nullable = true)
 |-- time: string (nullable = true)
 |-- serverip: string (nullable = true)
 |-- method: string (nullable = true)
 |-- uri: string (nullable = true)
 |-- querystirng: string (nullable = true)
 |-- port: integer (nullable = true)
 |-- username: string (nullable = true)
 |-- clientip: string (nullable = true)
 |-- useragent: string (nullable = true)
 |-- referrer: string (nullable = true)
 |-- statuscode: integer (nullable = true)
```

7. Let's handle the IP Address lookup data. Write spark code to load the **iplookup.json** file from Minio into the data frame **ips1**. Show the first 10 rows and print the schema, for a screenshot to include code and output.



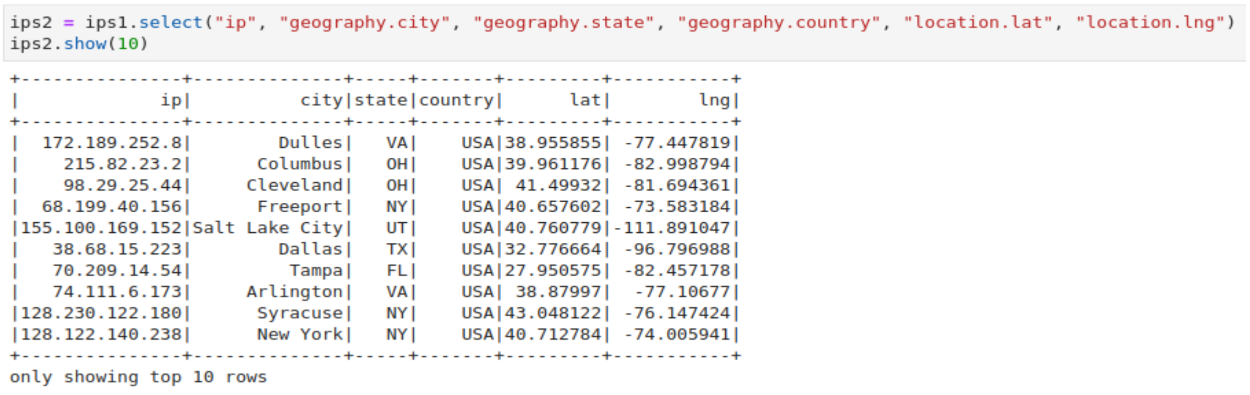
```
iplookup_in = f"s3a://{s3_bucket}/iplookup/iplookup.json"
ips1 = spark.read \
    .option("multiline", True) \
    .json(iplookup_in)
ips1.show(10)
```

```
+--------------------+---------------+--------------------+
|           geography|             ip|            location|
+--------------------+---------------+--------------------+
|   {Dulles, USA, VA}|  172.189.252.8|{38.955855, -77.4...|
| {Columbus, USA, OH}|    215.82.23.2|{39.961176, -82.9...|
|{Cleveland, USA, OH}|    98.29.25.44|{41.49932, -81.69...|
| {Freeport, USA, NY}|  68.199.40.156|{40.657602, -73.5...|
|{Salt Lake City, ...|155.100.169.152|{40.760779, -111....|
|   {Dallas, USA, TX}|   38.68.15.223|{32.776664, -96.7...|
|    {Tampa, USA, FL}|   70.209.14.54|{27.950575, -82.4...|
|{Arlington, USA, VA}|   74.111.6.173|{38.87997, -77.10...|
| {Syracuse, USA, NY}|128.230.122.180|{43.048122, -76.1...|
| {New York, USA, NY}|128.122.140.238|{40.712784, -74.0...|
+--------------------+---------------+--------------------+
only showing top 10 rows
```

```
ips1.printSchema()
```

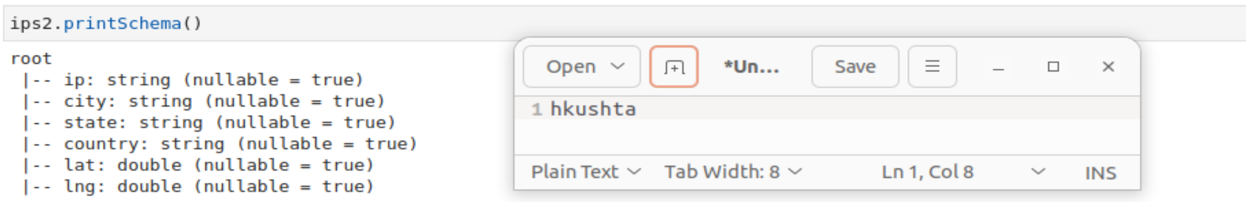
```
root
 |-- geography: struct (nullable = true)
 |    |-- city: string (nullable = true)
 |    |-- country: string (nullable = true)
 |    |-- state: string (nullable = true)
 |-- ip: string (nullable = true)
 |-- location: struct (nullable = true)
 |    |-- lat: double (nullable = true)
 |    |-- lng: double (nullable = true)
```

8. We need to flatten the nested JSON data, use the **select()** function with dot notation to do this, saving the dataframe as **ips2**. Provide a screenshot of the schema and output of the first few rows.



```
ips2 = ips1.select("ip", "geography.city", "geography.state", "geography.country", "location.lat", "location.lng")
ips2.show(10)
```

```
+---------------+-------------+-----+-------+---------+-----------+
|             ip|         city|state|country|      lat|        lng|
+---------------+-------------+-----+-------+---------+-----------+
|  172.189.252.8|       Dulles|   VA|    USA|38.955855| -77.447819|
|    215.82.23.2|     Columbus|   OH|    USA|39.961176| -82.998794|
|     98.29.25.44|    Cleveland|   OH|    USA| 41.49932| -81.694361|
|  68.199.40.156|     Freeport|   NY|    USA|40.657602| -73.583184|
|155.100.169.152|Salt Lake City|   UT|    USA|40.760779|-111.891047|
|    38.68.15.223|       Dallas|   TX|    USA|32.776664| -96.796988|
|    70.209.14.54|        Tampa|   FL|    USA|27.950575| -82.457178|
|    74.111.6.173|    Arlington|   VA|    USA| 38.87997|  -77.10677|
|128.230.122.180|     Syracuse|   NY|    USA|43.048122| -76.147424|
|128.122.140.238|     New York|   NY|    USA|40.712784| -74.005941|
+---------------+-------------+-----+-------+---------+-----------+
only showing top 10 rows
```

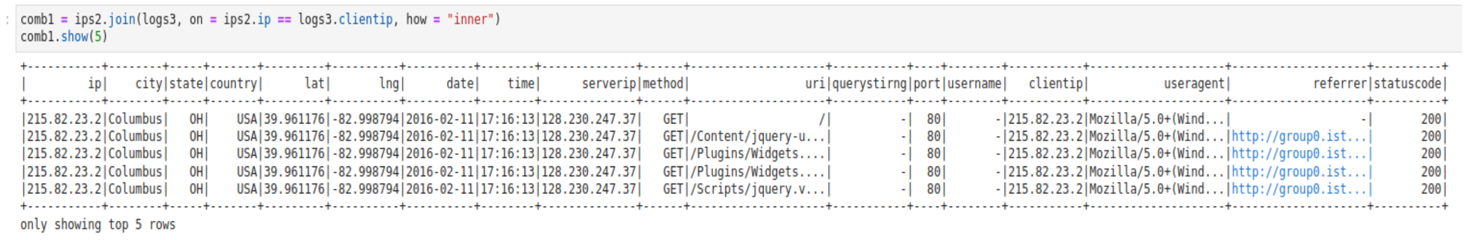


```
ips2.printSchema()
```

```
root
 |-- ip: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- country: string (nullable = true)
 |-- lat: double (nullable = true)
 |-- lng: double (nullable = true)
```

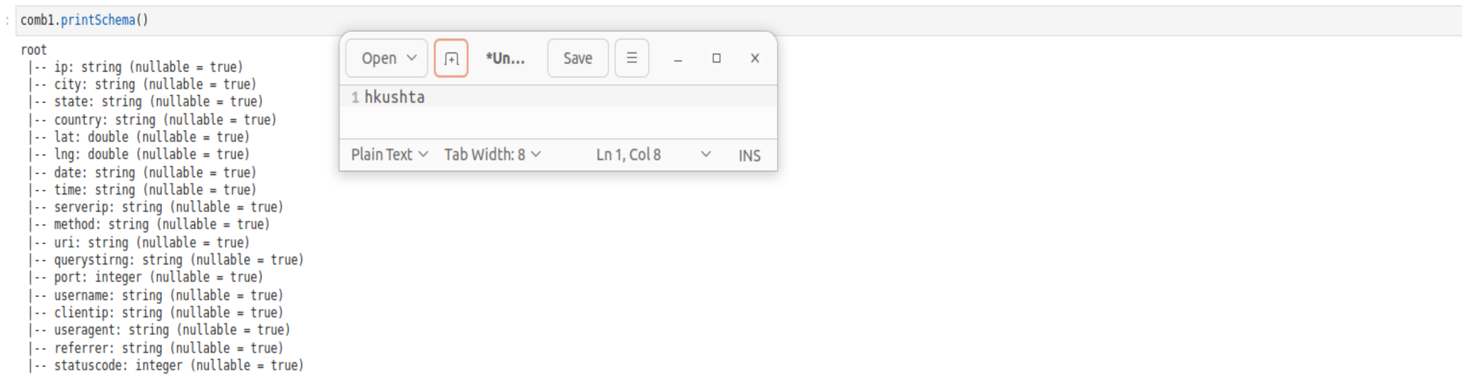
```
1 hkushta
```

9. Now join the two dataframe together on their business key, making the new dataframe **comb1.** Provide a schema and sample of the first few rows in your screenshot.



```
comb1 = ips2.join(logs3, on = ips2.ip == logs3.clientip, how = "inner")
comb1.show(5)
```

```
+-----------+--------+-----+-------+---------+----------+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|         ip|    city|state|country|      lat|       lng|      date|    time|      serverip|method|                 uri|querystirng|port|username|   clientip|           useragent|            referrer|statuscode|
+-----------+--------+-----+-------+---------+----------+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
|215.82.23.2|Columbus|   OH|    USA|39.961176|-82.998794|2016-02-11|17:16:13|128.230.247.37|   GET|                   /|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|                   -|       200|
|215.82.23.2|Columbus|   OH|    USA|39.961176|-82.998794|2016-02-11|17:16:13|128.230.247.37|   GET|/Content/jquery-u...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|215.82.23.2|Columbus|   OH|    USA|39.961176|-82.998794|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|215.82.23.2|Columbus|   OH|    USA|39.961176|-82.998794|2016-02-11|17:16:13|128.230.247.37|   GET|/Plugins/Widgets....|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
|215.82.23.2|Columbus|   OH|    USA|39.961176|-82.998794|2016-02-11|17:16:13|128.230.247.37|   GET|/Scripts/jquery.v...|          -|  80|       -|215.82.23.2|Mozilla/5.0+(Wind...|http://group0.ist...|       200|
+-----------+--------+-----+-------+---------+----------+----------+--------+--------------+------+--------------------+-----------+----+--------+-----------+--------------------+--------------------+----------+
only showing top 5 rows
```

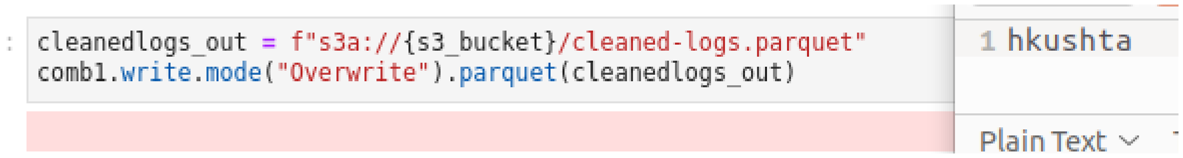


```
comb1.printSchema()
```

```
root
 |-- ip: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- country: string (nullable = true)
 |-- lat: double (nullable = true)
 |-- lng: double (nullable = true)
 |-- date: string (nullable = true)
 |-- time: string (nullable = true)
 |-- serverip: string (nullable = true)
 |-- method: string (nullable = true)
 |-- uri: string (nullable = true)
 |-- querystirng: string (nullable = true)
 |-- port: integer (nullable = true)
 |-- username: string (nullable = true)
 |-- clientip: string (nullable = true)
 |-- useragent: string (nullable = true)
 |-- referrer: string (nullable = true)
 |-- statuscode: integer (nullable = true)
```

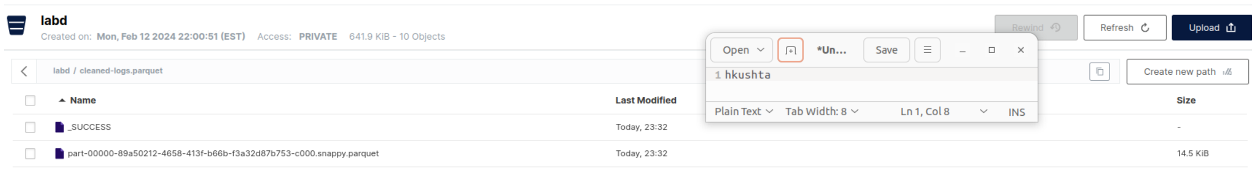
```
1 hkushta
```

10. Write the **comb1** Dataframe in **parquet** format back to Minio in the folder **cleaned-logs**. Again, show evidence the code ran, and the file was created.



```
cleanedlogs_out = f"s3a://{s3_bucket}/cleaned-logs.parquet"
comb1.write.mode("Overwrite").parquet(cleanedlogs_out)
```

```
hkushta
```



labd
Created on: Mon, Feb 12 2024 22:00:51 (EST) Access: PRIVATE 641.9 KiB - 10 Objects

labd / cleaned-logs.parquet

Name	Last Modified	Size
_SUCCESS	Today, 23:32	-
part-00000-89a50212-4658-413f-b66b-f3a32d87b753-c000.snappy.parquet	Today, 23:32	14.5 KiB

```
hkushta
```