

CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB ASSIGNMENT - 5

Banner ID: B00948977

Git Assignment Link :

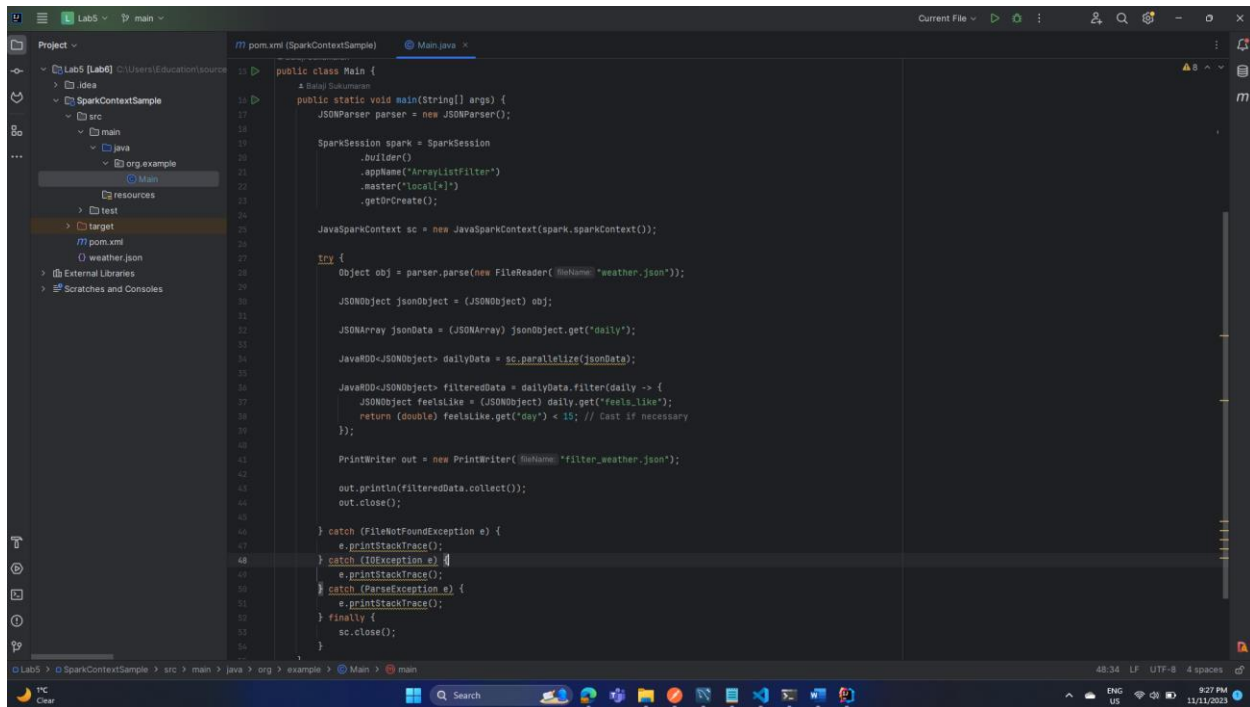
https://git.cs.dal.ca/sukumaran/csci5408_f23_b00948977_balaji_sukumaran

Table of contents

Problem Statement 1: Create a java program to find 5 days weather data.....	1
Problem Statement 2: Create an apache spark instance in GCP.....	2
Problem Statement 3: Execute the Java program along with its dependency.....	3

Problem Statement 1: Create a java program to find 5 days weather data

The below java program will read the weather.json file and extract the data which whose feels like temperature is less than 5 C. It uses the spark context to filter the daily weather object.



```
1 public class Main {
2     public static void main(String[] args) {
3         JSONParser parser = new JSONParser();
4
5         SparkSession spark = SparkSession
6             .builder()
7             .appName("ArraylistFilter")
8             .master("local[*]")
9             .getOrCreate();
10
11         JavaSparkContext sc = new JavaSparkContext(spark.sparkContext());
12
13         try {
14             Object obj = parser.parse(new FileReader("file://weather.json"));
15
16             JSONObject jsonObject = (JSONObject) obj;
17
18             JSONArray jsonData = (JSONArray) jsonObject.get("daily");
19
20             JavaRDD<JSONObject> dailyData = sc.parallelize(jsonData);
21
22             JavaRDD<JSONObject> filteredData = dailyData.filter(daily -> {
23                 JSONObject feelsLike = (JSONObject) daily.get("feels_like");
24                 return (double) feelsLike.get("day") < 15; // Cast if necessary
25             });
26
27             PrintWriter out = new PrintWriter("file://filter_weather.json");
28
29             out.println(filteredData.collect());
30             out.close();
31
32         } catch (FileNotFoundException e) {
33             e.printStackTrace();
34         } catch (IOException e) {
35             e.printStackTrace();
36         } catch (ParseException e) {
37             e.printStackTrace();
38         } finally {
39             sc.close();
40         }
41     }
42 }
```

Figure 1: java program for parsing the weather file

Problem Statement 2: Create an apache spark instance in GCP

Created an apache spark instance in the Google cloud platform to run the java file.

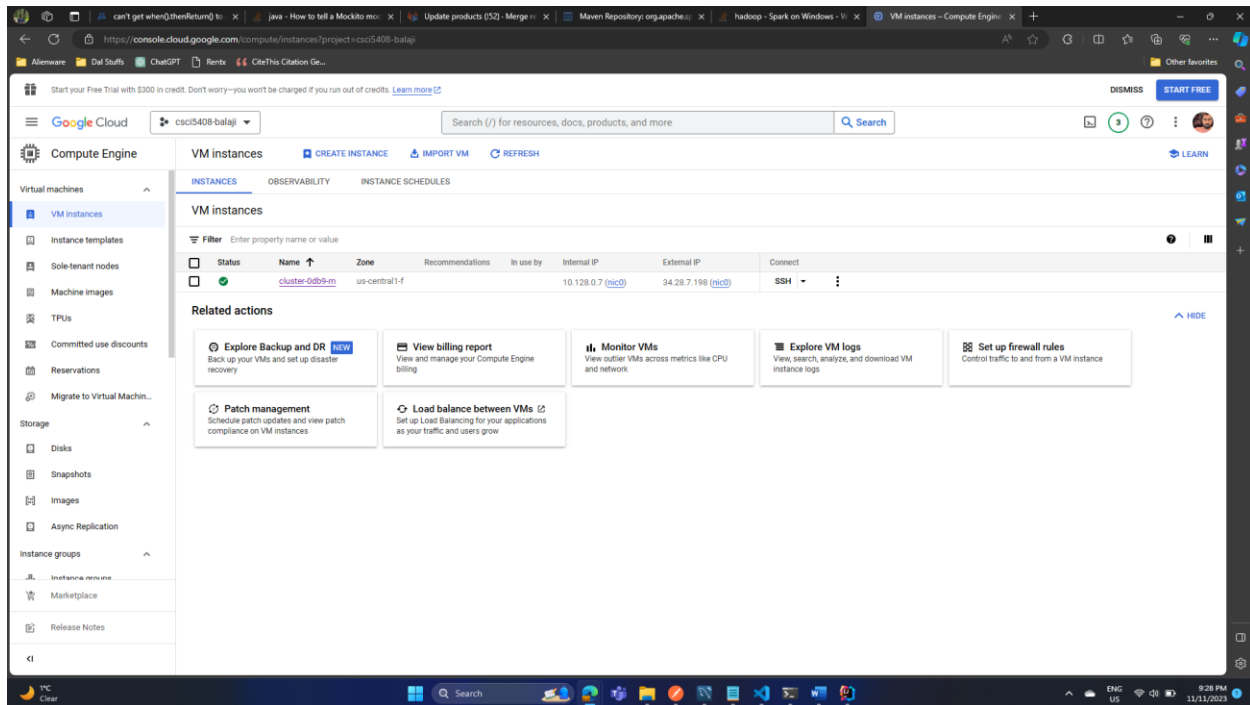


Figure 2: apache spark instance

Problem Statement 3: Execute the Java program along with its dependency JAR files in the created apache spark cluster

Step 1: Uploaded the following java program JAR and json-simple-1.1.1.jar and weather.json

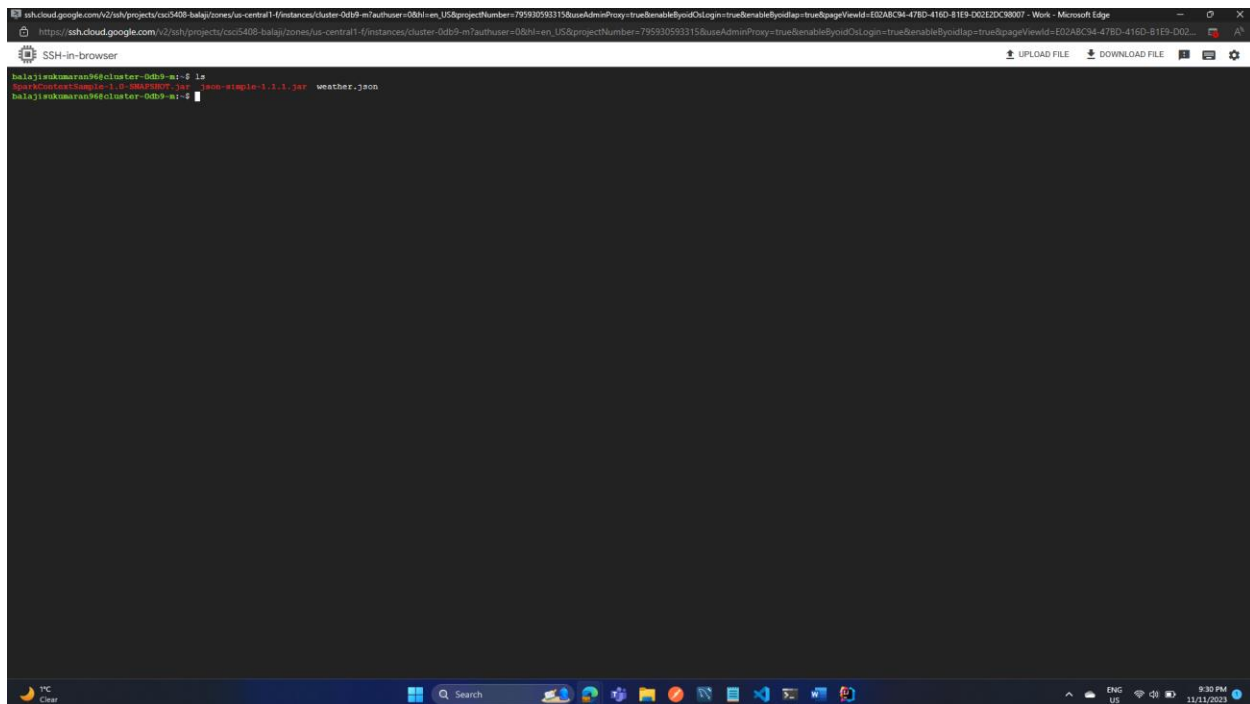


Figure 3: cluster contents

Step 2: Execute the JAR file with the dependency JARs using this command “spark-submit --jars ./json-simple-1.1.1.jar --class org.example.Main ./SparkContextSample-1.0-SNAPSHOT.jar”

Executed successfully.

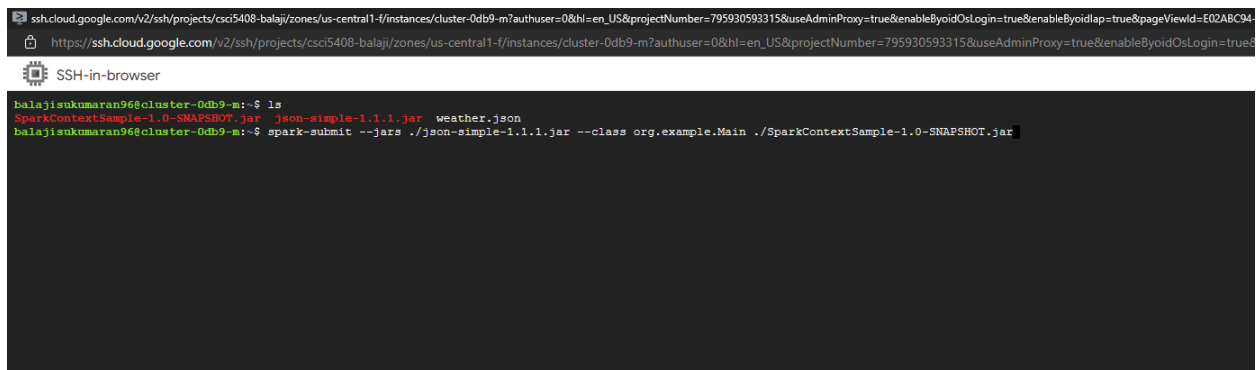
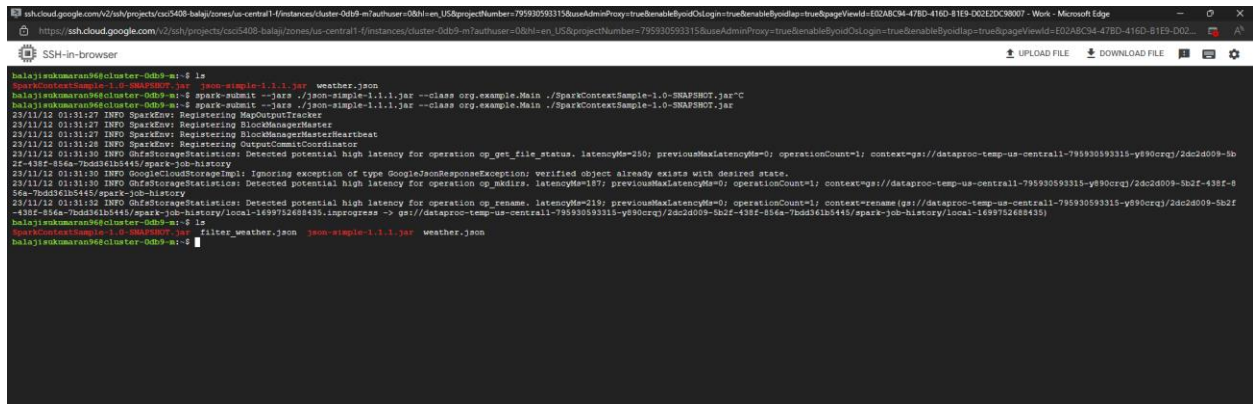


Figure 4: executed successfully

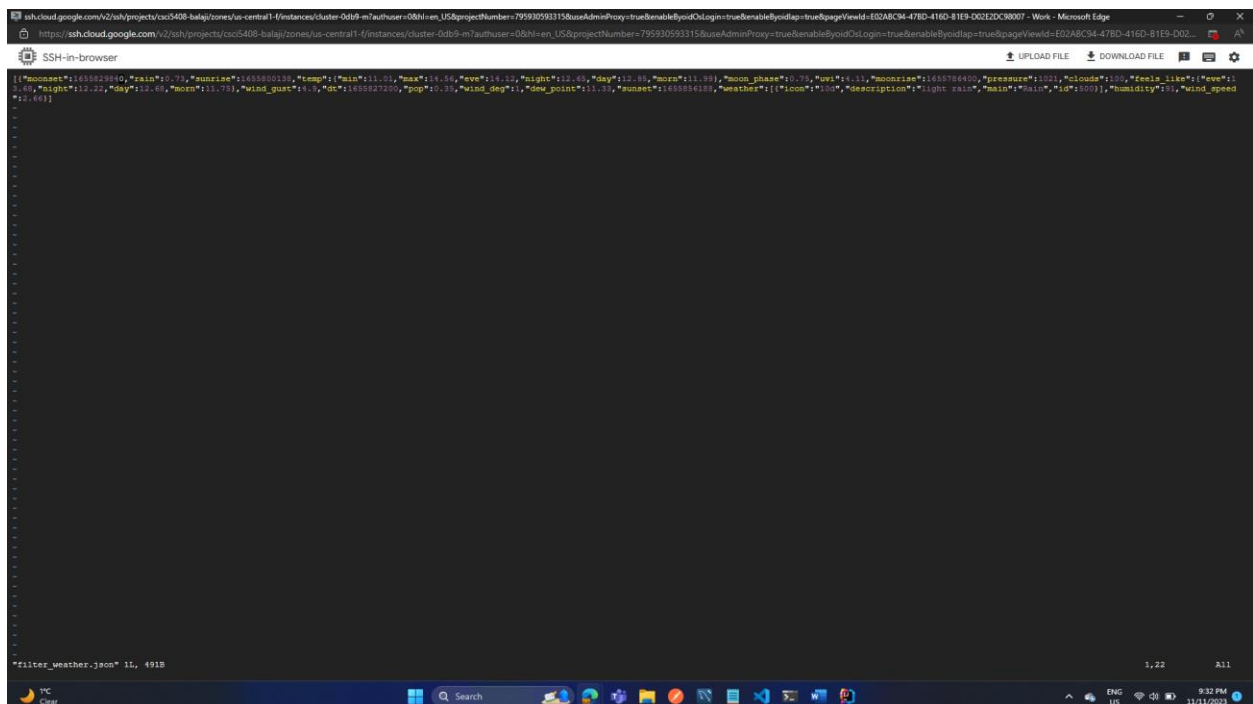
Step 3: Following is the generated output file.



```
hala@jaskumar94ec1aster-0d9-m:~$ ls
SparkContextSample-1.0-SNAPSHOT.jar  json-simple-1.1.1.jar  weather.json
hala@jaskumar94ec1aster-0d9-m:~$ spark-submit --jars ./json-simple-1.1.1.jar --class org.example.Main ./SparkContextSample-1.0-SNAPSHOT.jar
23/11/12 01:31:27 INFO SparkEnv: Registering MapOutputTracker
23/11/12 01:31:27 INFO SparkEnv: Registering BlockManagerMaster
23/11/12 01:31:27 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/11/12 01:31:28 INFO SparkEnv: Registering OutputCommitCoordinator
23/11/12 01:31:30 INFO GFileStorageStatistics: Detected potential high latency for operation op_get_file_status. latencyMs=250; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-795930593315-y890crqj/2dc2d009-bb2f-438f-856a-7bdd361b5445/spark-job-history
23/11/12 01:31:30 INFO GoogleCloudStorageTempl: Ignoring exception of type GoogleJsonResponseException: verified object already exists with desired state.
23/11/12 01:31:30 INFO GFileStorageStatistics: Detected potential high latency for operation op_mkdirs. latencyMs=187; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-795930593315-y890crqj/2dc2d009-bb2f-438f-856a-7bdd361b5445/spark-job-history
23/11/12 01:31:32 INFO GFileStorageStatistics: Detected potential high latency for operation op_rename. latencyMs=219; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-795930593315-y890crqj/2dc2d009-bb2f-438f-856a-7bdd361b5445/spark-job-history/local-1699752688435.improgrss -> gs://dataproc-temp-us-central1-795930593315-y890crqj/2dc2d009-bb2f-438f-856a-7bdd361b5445/spark-job-history/local-1699752688435
hala@jaskumar94ec1aster-0d9-m:~$ ls
SparkContextSample-1.0-SNAPSHOT.jar  filter_weather.json  json-simple-1.1.1.jar  weather.json
hala@jaskumar94ec1aster-0d9-m:~$
```

Figure 5: output file is generated

Following is the output file.



```
hala@jaskumar94ec1aster-0d9-m:~$ cat filter_weather.json
[[{"moonset":1688829810,"moon":10.75,"moonrise":1688880030,"temp":{"min":11.02,"max":14.46,"ave":13.42,"night":12.85,"day":12.85,"morn":11.38},"moon_phase":10.75,"sun":{"min":1688880030,"pressure":1021,"clouds":100,"feels_like":{"vev":13.68,"night":12.22,"day":12.48,"morn":11.75},"wind_gust":14.7,"dt":1688827200,"pop":0.35,"wind_deg":11,"dew_point":11.33,"sunet":1688856189,"weather":[{"icon":"10n","description":"light rain","main":"Rain","id":8009}],"humidity":79,"wind_speed":12.69}]]
hala@jaskumar94ec1aster-0d9-m:~$
```

Figure 6: output file contents

REFERRRENCES:

- [1] "Apache Hadoop," IBM. [Online]. Available: <https://www.ibm.com/analytics/hadoop>. [Accessed: 21-Oct-2023].
- [2] "Spark SQL, DataFrames and datasets guide," Spark SQL and DataFrames - Spark3.0.1 Documentation. [Online]. Available: <https://spark.apache.org/docs/3.0.1/sql-programming-guide.html>. [Accessed: 21-Oct-2023].