CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB ASSIGNMENT - 5

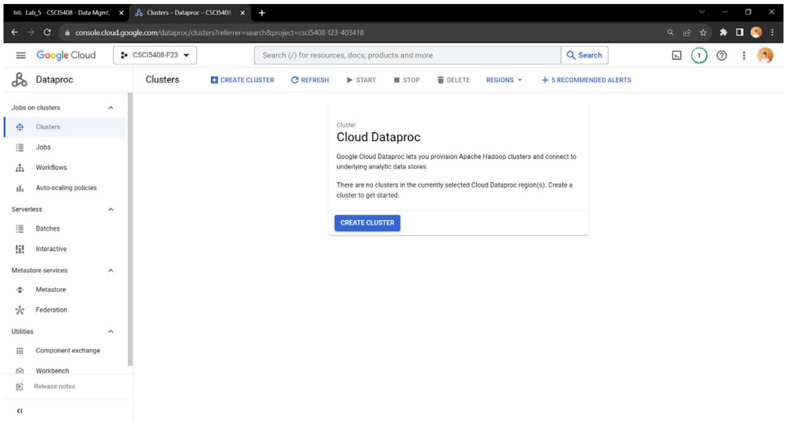
Banner ID: B00952865

GitLab Assignment Link: https://git.cs.dal.ca/apurohit/CSCI5408_F23_B00952865_AdityaMaheshbhai_Purohit/-/tree/main/Lab5

Table of Contents

Spark Set-up

Step-1: Goto Cloud Dataproc service in Google cloud platform using the search option and click create cluster.



console.cloud.google.com/dataproc/clusters?referrer=search&project=csci5408-f23-403418

Search (/) for resources, docs, products and more

Cluster

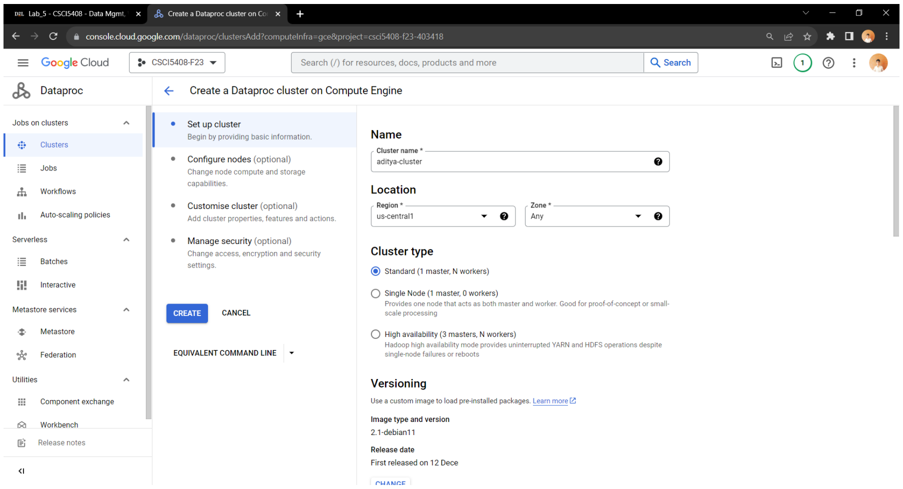
Cloud Dataproc

Google Cloud Dataproc lets you provision Apache Hadoop clusters and connect to underlying analytic data stores.

There are no clusters in the currently selected Cloud Dataproc region(s). Create a cluster to get started.

Figure 1: GCP Dataproc homescreen [1].

Step-2: Give any convenient name to the cluster.



console.cloud.google.com/dataproc/clustersAdd?computeInfra=gce&project=csci5408-f23-403418

Create a Dataproc cluster on Compute Engine

- Set up cluster
  Begin by providing basic information.
- Configure nodes (optional)
  Change node compute and storage capabilities.
- Customise cluster (optional)
  Add cluster properties, features and actions.
- Manage security (optional)
  Change access, encryption and security settings.

Name

Cluster name * aditya-cluster

Location

Region * us-central1

Zone * Any

Cluster type

● Standard (1 master, N workers)

○ Single Node (1 master, 0 workers)
Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing

○ High availability (3 masters, N workers)
Hadoop high availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Versioning

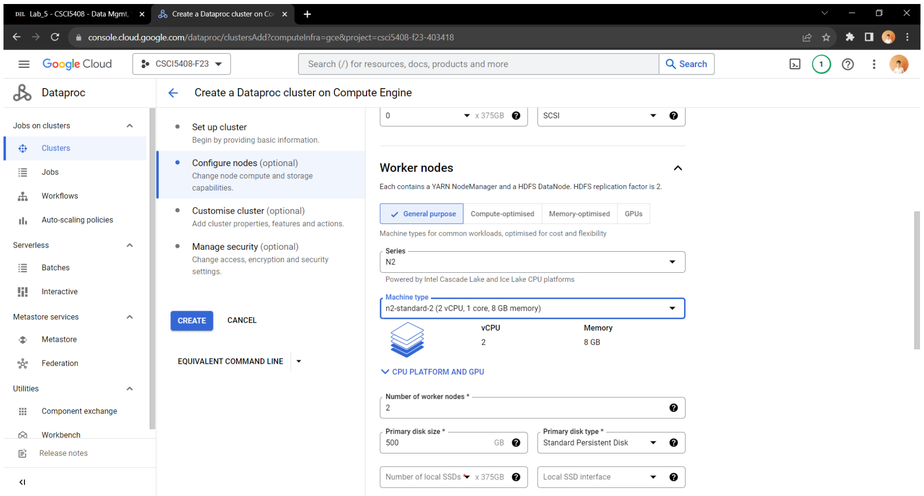
Use a custom image to load pre-installed packages. Learn more

Image type and version
2.1-debian11

Release date
First released on 12 Dece

Figure 2: Creating cluster [1].

Step-3: Reduced the vCPUs of worked nodes to 2, due to usage limits of my account.


Lab_5 - CSCI5408 - Data Mgmt.
Create a Dataproc cluster on Co
console.cloud.google.com/dataproc/clustersAdd?computeInfra=gce&project=csci5408-f23-403418
Google Cloud
CSCI5408-F23
Search (/) for resources, docs, products and more
Search
Dataproc
Jobs on clusters
Clusters
Jobs
Workflows
Auto-scaling policies
Serverless
Batches
Interactive
Metastore services
Metastore
Federation
Utilities
Component exchange
Workbench
Release notes
Create a Dataproc cluster on Compute Engine
Set up cluster
Begin by providing basic information.
Configure nodes (optional)
Change node compute and storage capabilities.
Customise cluster (optional)
Add cluster properties, features and actions.
Manage security (optional)
Change access, encryption and security settings.
CREATE
CANCEL
EQUIVALENT COMMAND LINE
General purpose
Compute-optimised
Memory-optimised
GPUs
CPU PLATFORM AND GPU

0 x 375GB

SCSI

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode. HDFS replication factor is 2.

Machine types for common workloads, optimised for cost and flexibility

Series: N2

Powered by Intel Cascade Lake and Ice Lake CPU platforms

Machine type: n2-standard-2 (2 vCPU, 1 core, 8 GB memory)

vCPU	Memory
2	8 GB

Number of worker nodes *: 2

Primary disk size *: 500 GB

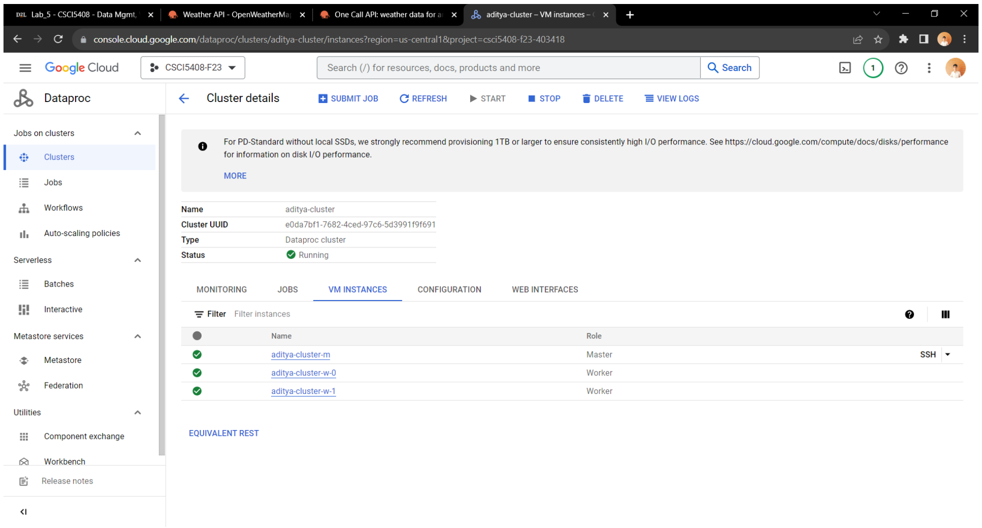
Primary disk type *: Standard Persistent Disk

Number of local SSDs * x 375GB

Local SSD interface

Figure 3: Worker nodes vCPU reduced [1].

Step-4: After a while, you should see all the nodes of cluster running.



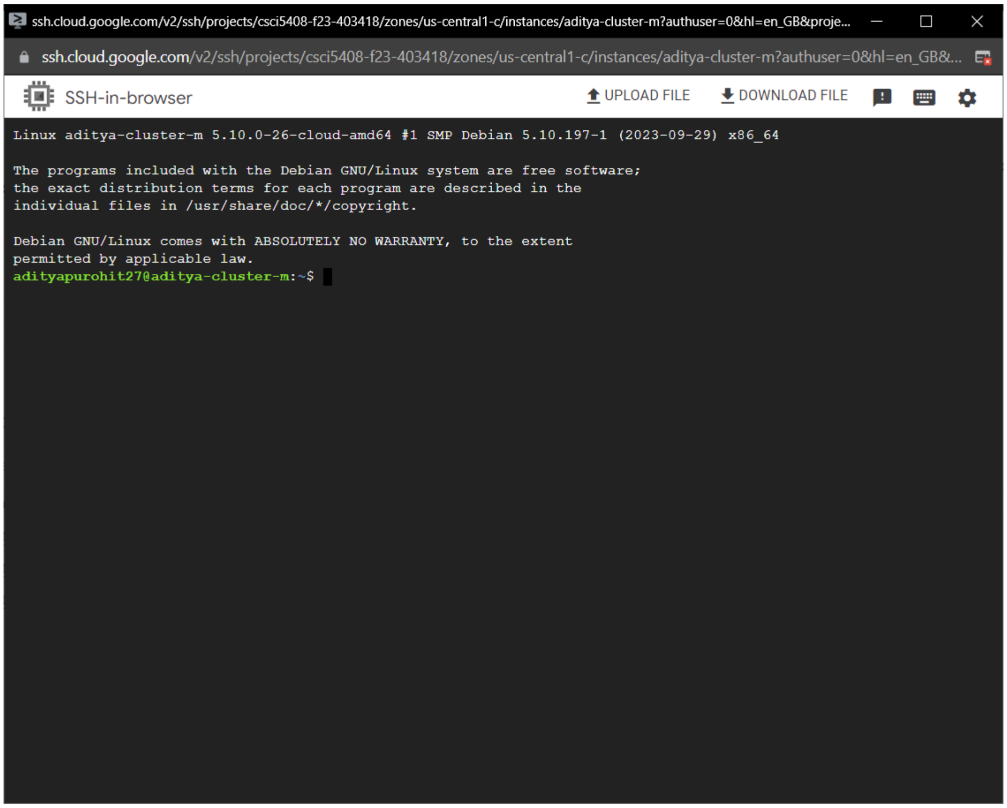
For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.google.com/compute/docs/disks/performance for information on disk I/O performance.

Name	aditya-cluster
Cluster UUID	e0da7bf1-7682-4ced-97c6-5d3991f9f691
Type	Dataproc cluster
Status	✓ Running

●	Name	Role
✓	aditya-cluster-m	Master
✓	aditya-cluster-w-0	Worker
✓	aditya-cluster-w-1	Worker

Figure 4: Cluster running [1].

Step-5: Connect to master node using ssh.

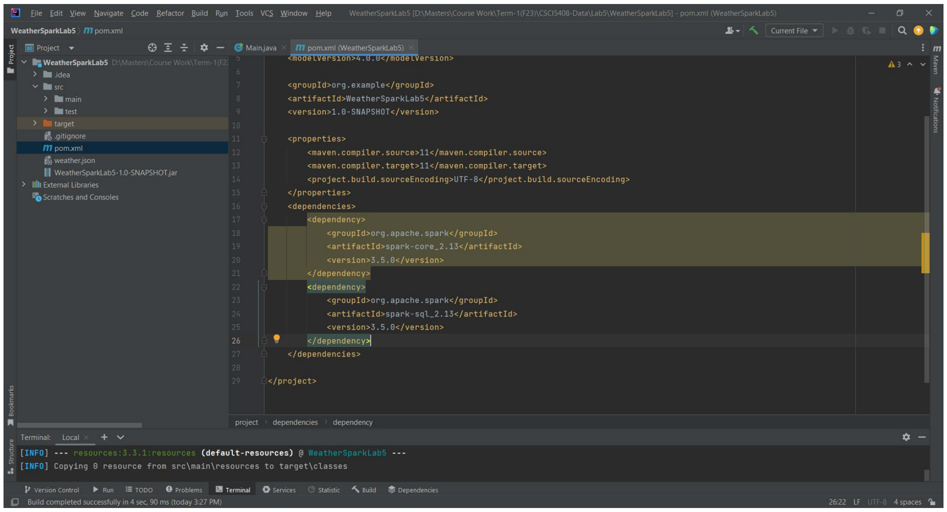

ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&proje...
ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
Linux aditya-cluster-m 5.10.0-26-cloud-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
adityapurohit27@aditya-cluster-m:~$ 
```

Figure 5: Connected to master node using SSH [1].

Java Program

Firstly, I have added 2 spark dependencies (core and sql) so that I can use them in my java program.



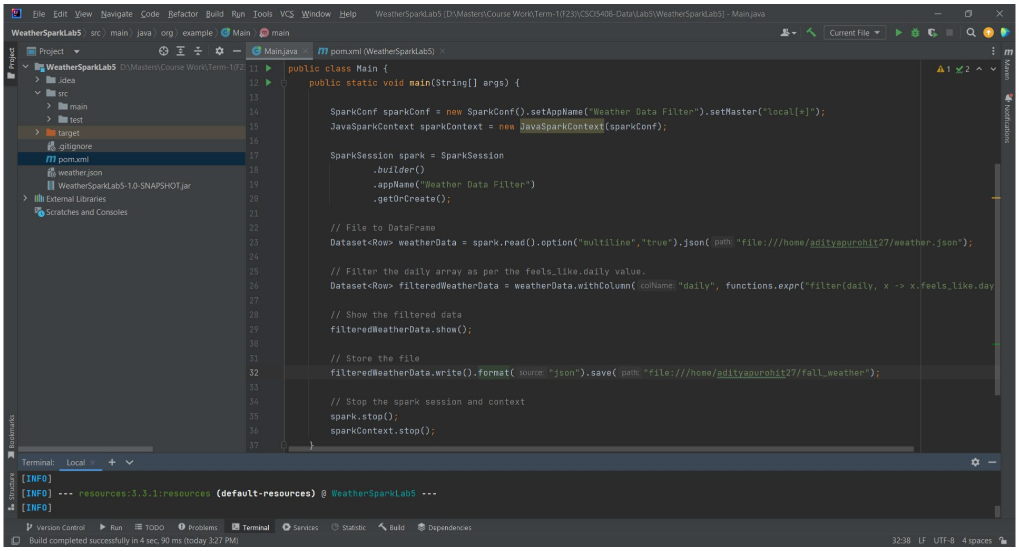
```xml
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>WeatherSparkLab5</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.13</artifactId>
            <version>3.5.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-sql_2.13</artifactId>
            <version>3.5.0</version>
        </dependency>
    </dependencies>

</project>
```

```
[INFO] --- resources:3.3.1:resources (default-resources) @ WeatherSparkLab5 ---
[INFO] Copying 0 resource from src\main\resources to target\classes
```

Build completed successfully in 4 sec, 90 ms (today 3:27 PM)

Figure 6: Adding spark dependencies in pom.xml [2] [3] [4] [5].

After that, I wrote the below program to read the weather.json file, filter the data where feels_like during day is less than 15. Exclusion of current, minutely, and hourly fields was not needed as they were not present in the original weather.json file.



```java
public class Main {
    public static void main(String[] args) {

        SparkConf sparkConf = new SparkConf().setAppName("Weather Data Filter").setMaster("local[*]");
        JavaSparkContext sparkContext = new JavaSparkContext(sparkConf);

        SparkSession spark = SparkSession
                .builder()
                .appName("Weather Data Filter")
                .getOrCreate();

        // File to DataFrame
        Dataset<Row> weatherData = spark.read().option("multiline","true").json("file:///home/adityapurohit27/weather.json");

        // Filter the daily array as per the feels_like.daily value.
        Dataset<Row> filteredWeatherData = weatherData.withColumn("daily", functions.expr("filter(daily, x -> x.feels_like.day

        // Show the filtered data
        filteredWeatherData.show();

        // Store the file
        filteredWeatherData.write().format("json").save("file:///home/adityapurohit27/fall_weather");

        // Stop the spark session and context
        spark.stop();
        sparkContext.stop();
    }
```

```
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ WeatherSparkLab5 ---
[INFO]
```

Build completed successfully in 4 sec, 90 ms (today 3:27 PM)

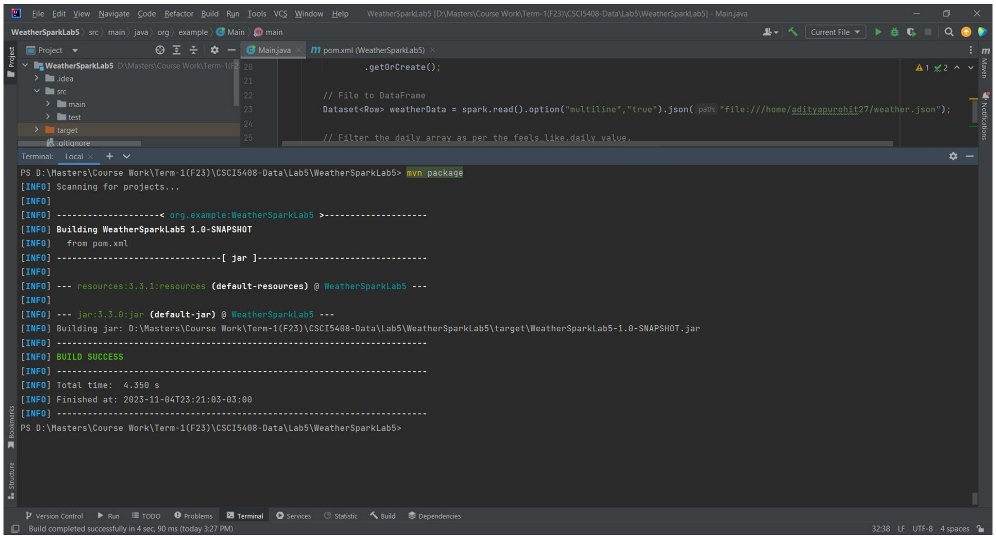
Figure 7: Overview of the java program that uses spark dependencies [2] [3] [4] [5] [6].

In the program I have firstly set the JavaSparkContext and started the SparkSession so that spark can do the further processing. Then, I loaded the weather.json from the master node into a spark

DataSet. After that, I used functions.expr of spark sql, to filter the “daily” array inside the weather.json. x inside the expr represents each json object inside the “daily” array and the x.feels_like.day is use to access the day temperature and check it with the 15 with less than operator to get only the needed entries from the “daily” array. This filter the daily array. Lastly, the filtered dataset it saved in a json under the fall_weather directory and then the spark session and context are closed [7] [8].

Execution of Java Program

Step-1: Run mvn package command to create a .jar file of java program.



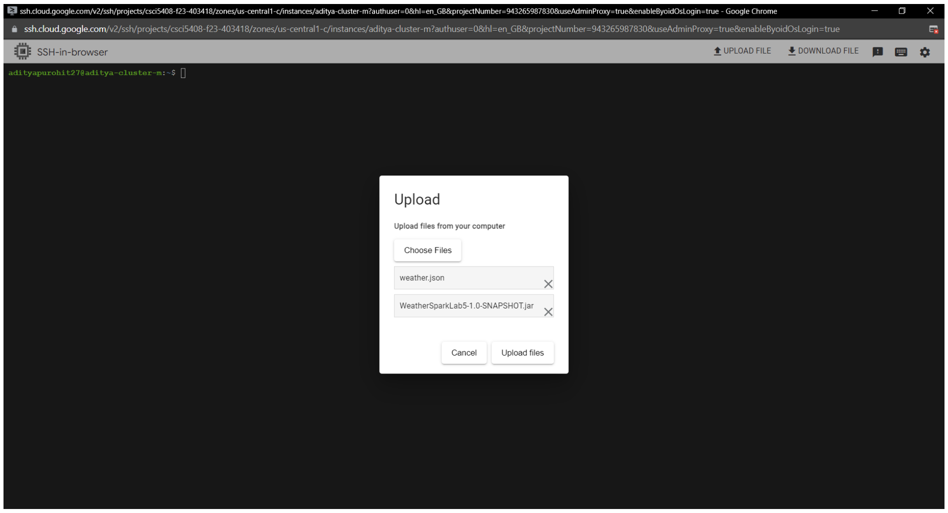
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
WeatherSparkLab5 [D:\Masters\Course Work\Term-1(F23)\CSCI5408-Data\Lab5\WeatherSparkLab5] - Main.java
WeatherSparkLab5 src main java org example Main main
Current File
Project
Main.java
pom.xml (WeatherSparkLab5)
WeatherSparkLab5 D:\Masters\Course Work\Term-1(F2
.idea
src
main
test
target
.gitignore
Terminal: Local
Version Control
Run
TODO
Problems
Terminal
Services
Statistic
Build
Dependencies
Build completed successfully in 4 sec, 90 ms (today 3:27 PM)
32:38 LF UTF-8 4 spaces

```
                .getOrCreate();

        // File to DataFrame
        Dataset<Row> weatherData = spark.read().option("multiline","true").json( path: "file:///home/adityapurohit27/weather.json");

        // Filter the daily array as per the feels_like.daily value.
```

```
PS D:\Masters\Course Work\Term-1(F23)\CSCI5408-Data\Lab5\WeatherSparkLab5> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< org.example:WeatherSparkLab5 >--------------------
[INFO] Building WeatherSparkLab5 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ WeatherSparkLab5 ---
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ WeatherSparkLab5 ---
[INFO] Building jar: D:\Masters\Course Work\Term-1(F23)\CSCI5408-Data\Lab5\WeatherSparkLab5\target\WeatherSparkLab5-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.350 s
[INFO] Finished at: 2023-11-04T23:21:03-03:00
[INFO] ------------------------------------------------------------------------
PS D:\Masters\Course Work\Term-1(F23)\CSCI5408-Data\Lab5\WeatherSparkLab5>
```

Figure 8: Build the jar file [2] [3] [4] [5].

Step-2: Upload the jar file and weather.json file (from the teams channel) into the master node, using the ssh window.



ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Upload
Upload files from your computer
Choose Files
Cancel
Upload files

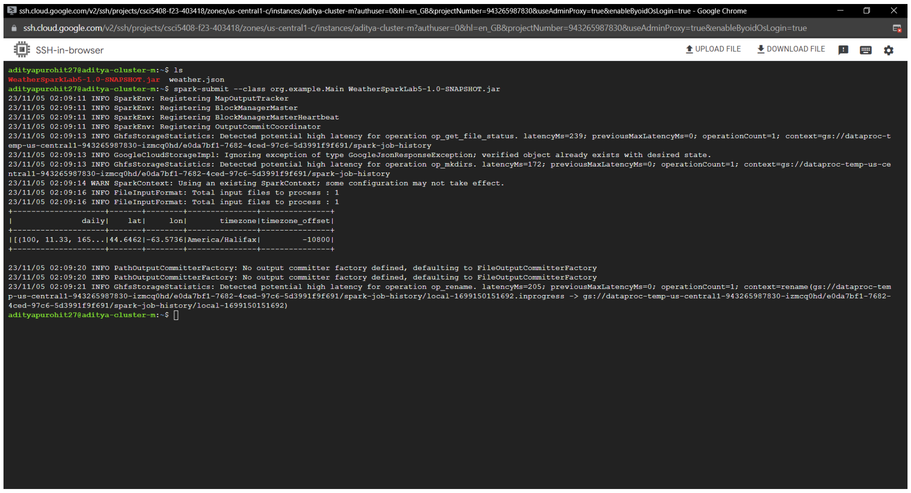
```
adityapurohit27@aditya-cluster-m:~$
```

weather.json

WeatherSparkLab5-1.0-SNAPSHOT.jar

Figure 9: Upload the JAR and JSON [1].

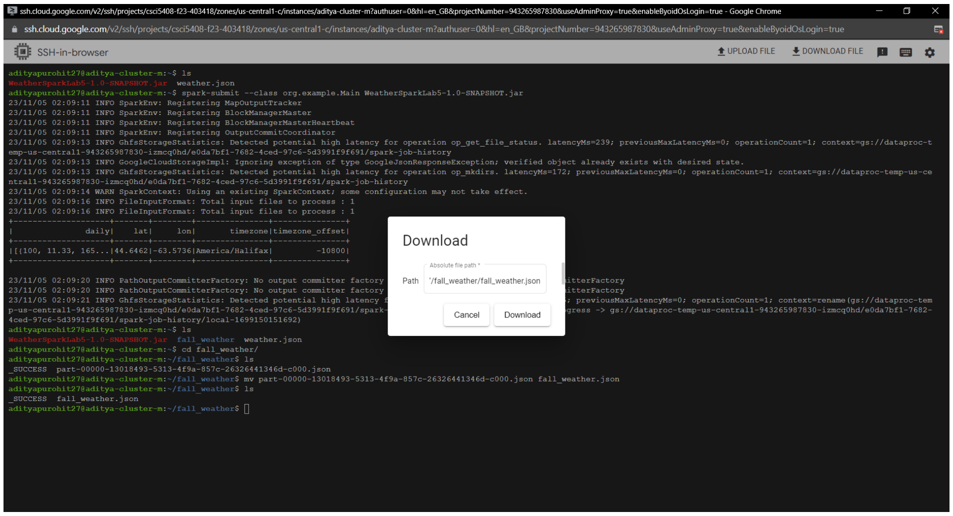
Step-3: Run the jar file using the spark command: “spark-submit --class org.example.Main WeatherSparkLab5-1.0-SNAPSHOT.jar”. Here the –class tag specifies the starting point of your java program.


ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
adityapurohit27@aditya-cluster-m:~$ ls
WeatherSparkLab5-1.0-SNAPSHOT.jar  weather.json
adityapurohit27@aditya-cluster-m:~$ spark-submit --class org.example.Main WeatherSparkLab5-1.0-SNAPSHOT.jar
23/11/05 02:09:11 INFO SparkEnv: Registering MapOutputTracker
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMaster
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/11/05 02:09:11 INFO SparkEnv: Registering OutputCommitCoordinator
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_get_file_status. latencyMs=239; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-t
emp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:13 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_mkdirs. latencyMs=172; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-ce
ntral1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:14 WARN SparkContext: Using an existing SparkContext; some configuration may not take effect.
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
+--------------------+-------+--------+---------------+---------------+
|               daily|    lat|     lon|       timezone|timezone_offset|
+--------------------+-------+--------+---------------+---------------+
|[{100, 11.33, 165...|44.6462|-63.5736|America/Halifax|         -10800|
+--------------------+-------+--------+---------------+---------------+

23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
23/11/05 02:09:21 INFO GhfsStorageStatistics: Detected potential high latency for operation op_rename. latencyMs=205; previousMaxLatencyMs=0; operationCount=1; context=rename(gs://dataproc-tem
p-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history/local-1699150151692.inprogress -> gs://dataproc-temp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-
4ced-97c6-5d3991f9f691/spark-job-history/local-1699150151692)
adityapurohit27@aditya-cluster-m:~$ 
```

Figure 10: Run the JAR file [1].

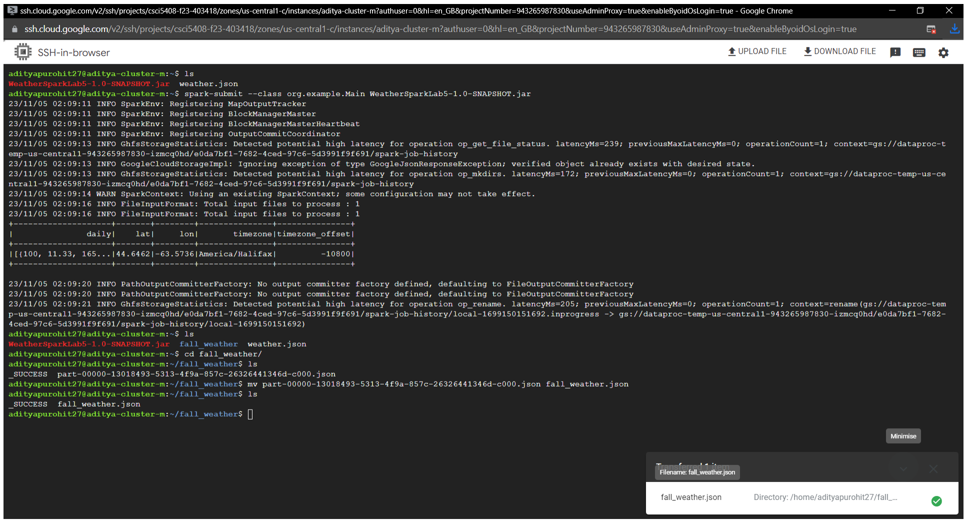
Step-4: Check the output json file under the fall_weather folder and download it and give it a suitable name using linux “mv” command.


ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Download
Absolute file path *
Cancel
Download

Path: '/fall_weather/fall_weather.json

```
adityapurohit27@aditya-cluster-m:~$ ls
WeatherSparkLab5-1.0-SNAPSHOT.jar  weather.json
adityapurohit27@aditya-cluster-m:~$ spark-submit --class org.example.Main WeatherSparkLab5-1.0-SNAPSHOT.jar
23/11/05 02:09:11 INFO SparkEnv: Registering MapOutputTracker
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMaster
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/11/05 02:09:11 INFO SparkEnv: Registering OutputCommitCoordinator
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_get_file_status. latencyMs=239; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-t
emp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:13 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_mkdirs. latencyMs=172; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-ce
ntral1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:14 WARN SparkContext: Using an existing SparkContext; some configuration may not take effect.
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
+--------------------+-------+--------+---------------+---------------+
|               daily|    lat|     lon|       timezone|timezone_offset|
+--------------------+-------+--------+---------------+---------------+
|[{100, 11.33, 165...|44.6462|-63.5736|America/Halifax|         -10800|
+--------------------+-------+--------+---------------+---------------+

23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory                    itterFactory
23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory                    itterFactory
23/11/05 02:09:21 INFO GhfsStorageStatistics: Detected potential high latency f                   ; previousMaxLatencyMs=0; operationCount=1; context=rename(gs://dataproc-tem
p-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-                   gress -> gs://dataproc-temp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-
4ced-97c6-5d3991f9f691/spark-job-history/local-1699150151692)
adityapurohit27@aditya-cluster-m:~$ ls
WeatherSparkLab5-1.0-SNAPSHOT.jar  fall_weather  weather.json
adityapurohit27@aditya-cluster-m:~$ cd fall_weather/
adityapurohit27@aditya-cluster-m:~/fall_weather$ ls
_SUCCESS  part-00000-13018493-5313-4f9a-857c-26326441346d-c000.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ mv part-00000-13018493-5313-4f9a-857c-26326441346d-c000.json fall_weather.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ ls
_SUCCESS  fall_weather.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ 
```

Figure 11: Rename the output JSON and download it [1].



ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true - Google Chrome
ssh.cloud.google.com/v2/ssh/projects/csci5408-f23-403418/zones/us-central1-c/instances/aditya-cluster-m?authuser=0&hl=en_GB&projectNumber=943265987830&useAdminProxy=true&enableByoidOsLogin=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Minimise
Transferred 1 item
Filename: fall_weather.json
fall_weather.json
Directory: /home/adityapurohit27/fall_...

```
adityapurohit27@aditya-cluster-m:~$ ls
WeatherSparkLab5-1.0-SNAPSHOT.jar  weather.json
adityapurohit27@aditya-cluster-m:~$ spark-submit --class org.example.Main WeatherSparkLab5-1.0-SNAPSHOT.jar
23/11/05 02:09:11 INFO SparkEnv: Registering MapOutputTracker
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMaster
23/11/05 02:09:11 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/11/05 02:09:11 INFO SparkEnv: Registering OutputCommitCoordinator
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_get_file_status. latencyMs=239; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-t
emp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:13 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
23/11/05 02:09:13 INFO GhfsStorageStatistics: Detected potential high latency for operation op_mkdirs. latencyMs=172; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-ce
ntral1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history
23/11/05 02:09:14 WARN SparkContext: Using an existing SparkContext; some configuration may not take effect.
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
23/11/05 02:09:16 INFO FileInputFormat: Total input files to process : 1
+--------------------+-------+--------+--------------+---------------+
|               daily|    lat|     lon|      timezone|timezone_offset|
+--------------------+-------+--------+--------------+---------------+
|[{100, 11.33, 165...|44.6462|-63.5736|America/Halifax|         -10800|
+--------------------+-------+--------+--------------+---------------+

23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
23/11/05 02:09:20 INFO PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
23/11/05 02:09:21 INFO GhfsStorageStatistics: Detected potential high latency for operation op_rename. latencyMs=205; previousMaxLatencyMs=0; operationCount=1; context=rename(gs://dataproc-tem
p-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-4ced-97c6-5d3991f9f691/spark-job-history/local-1699150151692.inprogress -> gs://dataproc-temp-us-central1-943265987830-izmcq0hd/e0da7bf1-7682-
4ced-97c6-5d3991f9f691/spark-job-history/local-1699150151692)
adityapurohit27@aditya-cluster-m:~$ ls
WeatherSparkLab5-1.0-SNAPSHOT.jar  fall_weather  weather.json
adityapurohit27@aditya-cluster-m:~$ cd fall_weather/
adityapurohit27@aditya-cluster-m:~/fall_weather$ ls
_SUCCESS  part-00000-13018493-5313-4f9a-857c-26326441346d-c000.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ mv part-00000-13018493-5313-4f9a-857c-26326441346d-c000.json fall_weather.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ ls
_SUCCESS  fall_weather.json
adityapurohit27@aditya-cluster-m:~/fall_weather$ 
```

Figure 12: Successful download of the fall_weather.json file [1].

References:

[1] "Dataproc," *Google Cloud*. [Online]. Available: https://cloud.google.com/dataproc?hl=en. [Accessed: 05-Nov-2023].

[2] "IntelliJ IDEA – the leading Java and Kotlin IDE," *JetBrains*. [Online]. Available: https://www.jetbrains.com/idea/. [Accessed: 05-Nov-2023].

[3] B. Porter, J. van Zyl, and O. Lamy, "Welcome to Apache maven," *Apache.org*. [Online]. Available: https://maven.apache.org/. [Accessed: 05-Nov-2023].

[4] "spark-core," *Mvnrepository.com*. [Online]. Available: https://mvnrepository.com/artifact/org.apache.spark/spark-core. [Accessed: 05-Nov-2023].

[5] "spark-sql," *Mvnrepository.com*. [Online]. Available: https://mvnrepository.com/artifact/org.apache.spark/spark-sql. [Accessed: 05-Nov-2023].

[6] "Java | Oracle," *Java.com*. [Online]. Available: https://www.java.com/en/. [Accessed: 05-Nov-2023].

[7] "Getting Started," *Apache.org*. [Online]. Available: https://spark.apache.org/docs/latest/sql-getting-started.html. [Accessed: 05-Nov-2023].

[8] "Functions (spark 3.0.2 JavaDoc)," Apache.org, 16-Feb-2021. [Online]. Available: https://spark.apache.org/docs/3.0.2/api/java/org/apache/spark/sql/functions.html. [Accessed: 05-Nov-2023].