CSCI 5408

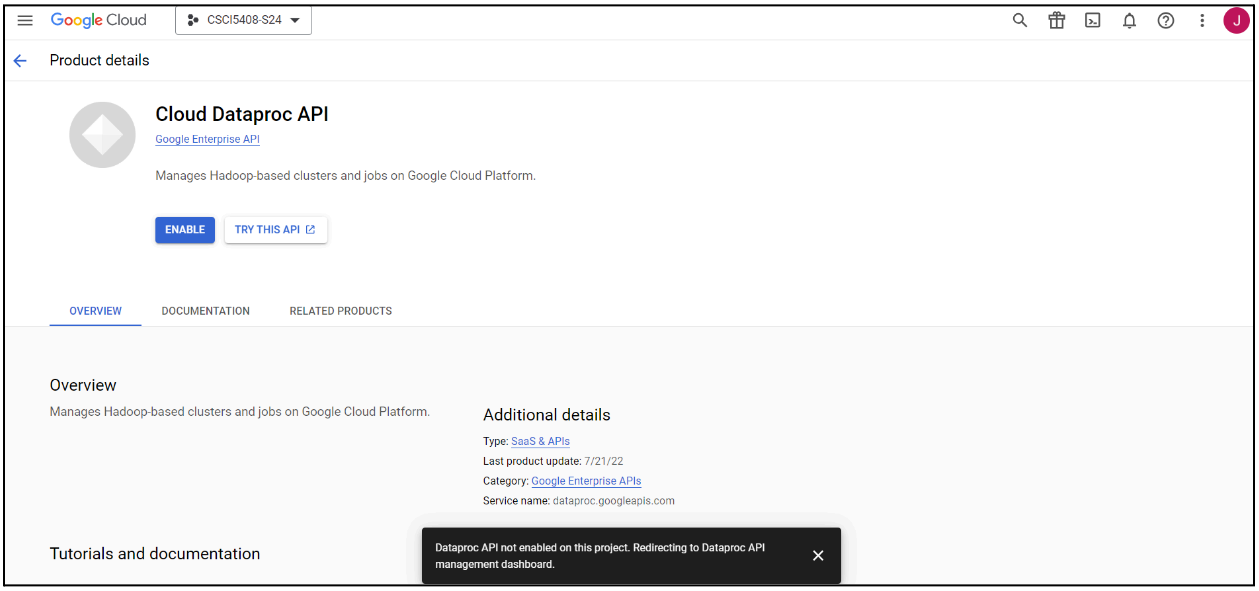
DATA MANAGEMENT AND WAREHOUSING

LAB-5: BIG DATA: HADOOP AND APACHE SPARK

Gitlab Link: https://git.cs.dal.ca/jspatel/csci5408_s24_b00982253_jay_patel.git

Table of Contents

1: Create Apache Spark Cluster on GCP



Cloud Dataproc API

Google Enterprise API

Manages Hadoop-based clusters and jobs on Google Cloud Platform.

Overview

Manages Hadoop-based clusters and jobs on Google Cloud Platform.

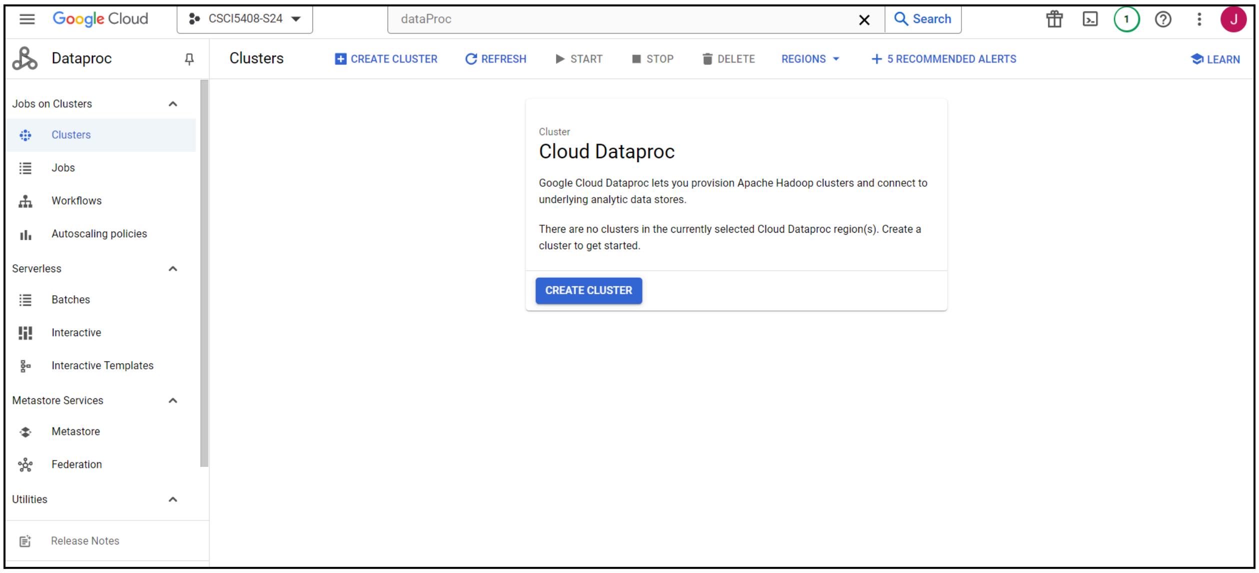
Additional details

Type: SaaS & APIs
Last product update: 7/21/22
Category: Google Enterprise APIs
Service name: dataproc.googleapis.com

Tutorials and documentation

Dataproc API not enabled on this project. Redirecting to Dataproc API management dashboard.

Figure 1: Enable API for Dataproc



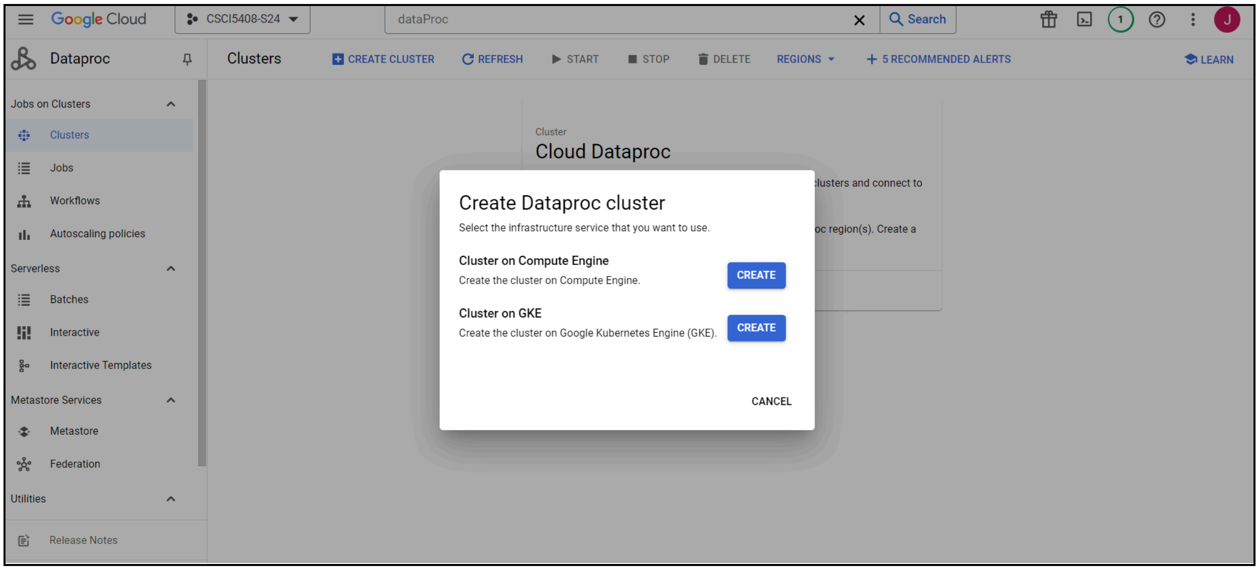
Cluster

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 2: Create Cluster



Create Dataproc cluster

Select the infrastructure service that you want to use.

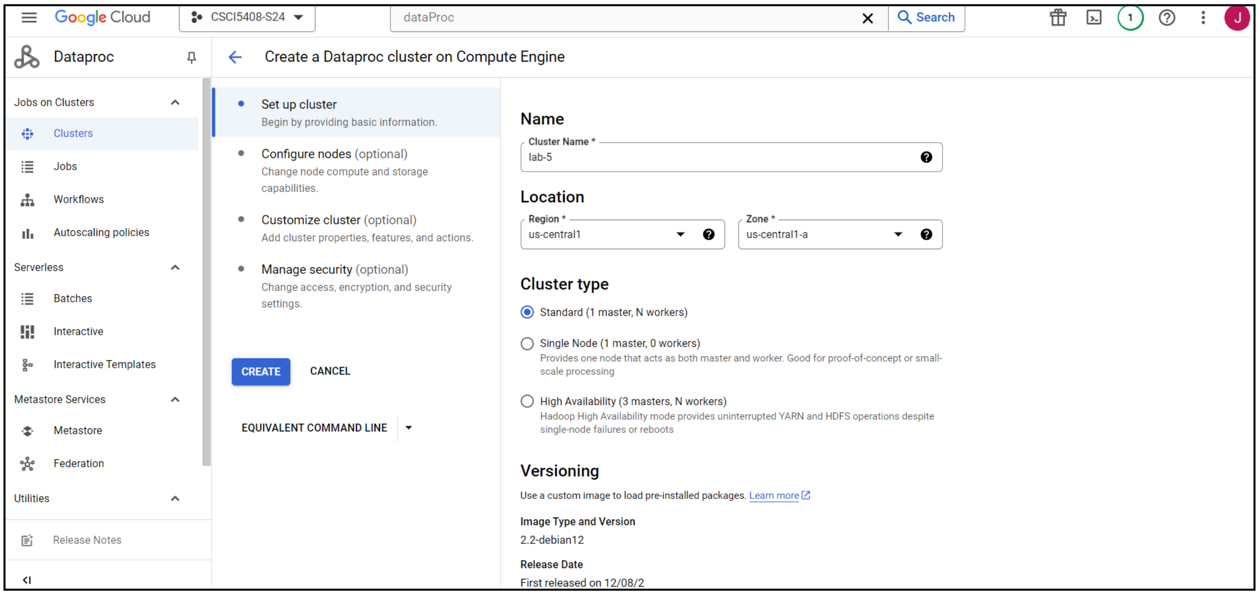
Cluster on Compute Engine

Create the cluster on Compute Engine.

Cluster on GKE

Create the cluster on Google Kubernetes Engine (GKE).

Figure 3: Select Compute Engine



Create a Dataproc cluster on Compute Engine

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

Name

Cluster Name * lab-5

Location

Region * us-central1

Zone * us-central1-a

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

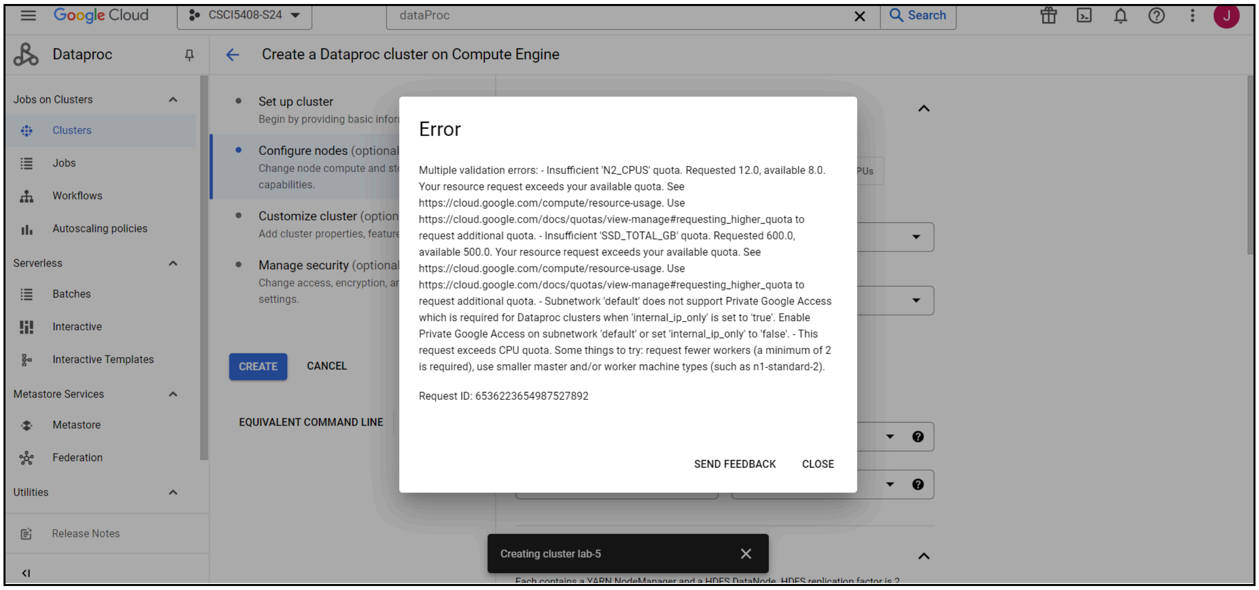
Image Type and Version

2.2-debian12

Release Date

First released on 12/08/2

Figure 4: Give a name, assign a location, and select cluster type.



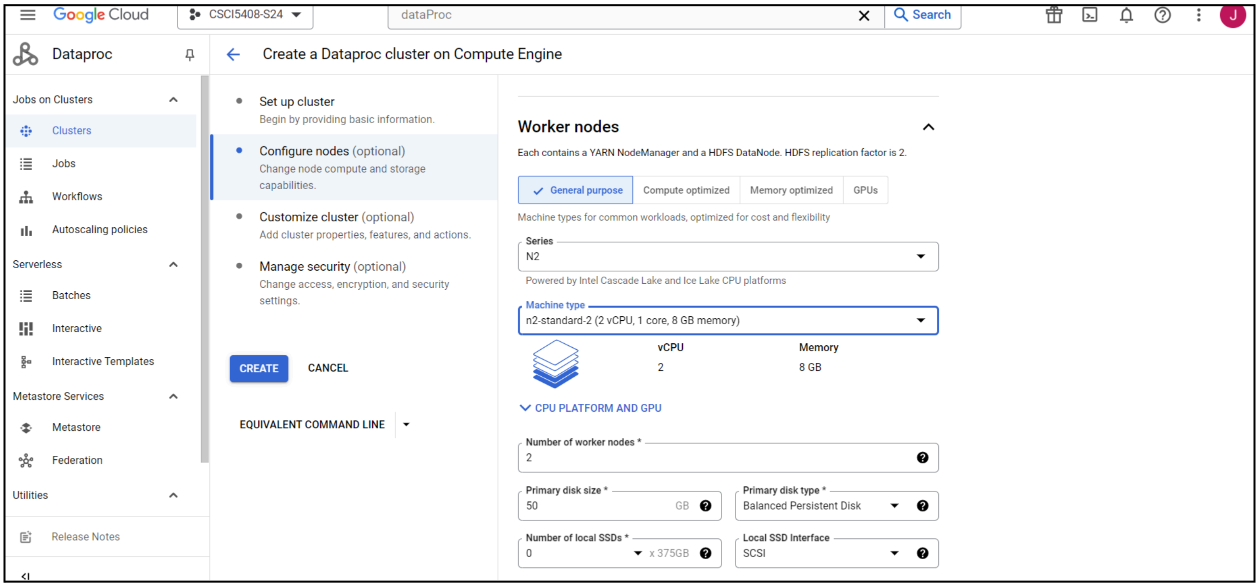
Error

Multiple validation errors: - Insufficient 'N2_CPUS' quota. Requested 12.0, available 8.0. Your resource request exceeds your available quota. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quotas/view-manage#requesting_higher_quota to request additional quota. - Insufficient 'SSD_TOTAL_GB' quota. Requested 600.0, available 500.0. Your resource request exceeds your available quota. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quotas/view-manage#requesting_higher_quota to request additional quota. - Subnetwork 'default' does not support Private Google Access which is required for Dataproc clusters when 'internal_ip_only' is set to 'true'. Enable Private Google Access on subnetwork 'default' or set 'internal_ip_only' to 'false'. - This request exceeds CPU quota. Some things to try: request fewer workers (a minimum of 2 is required), use smaller master and/or worker machine types (such as n1-standard-2).

Request ID: 6536223654987527892

Creating cluster lab-5

Figure 5: Error due to incorrect use of machine type for worker node.


Google Cloud
CSCI5408-S24
dataProc
Search
Dataproc
Create a Dataproc cluster on Compute Engine
Jobs on Clusters
Clusters
Jobs
Workflows
Autoscaling policies
Serverless
Batches
Interactive
Interactive Templates
Metastore Services
Metastore
Federation
Utilities
Release Notes
Set up cluster
Begin by providing basic information.
Configure nodes (optional)
Change node compute and storage capabilities.
Customize cluster (optional)
Add cluster properties, features, and actions.
Manage security (optional)
Change access, encryption, and security settings.
CREATE
CANCEL
EQUIVALENT COMMAND LINE

Worker nodes

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

Machine types for common workloads, optimized 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

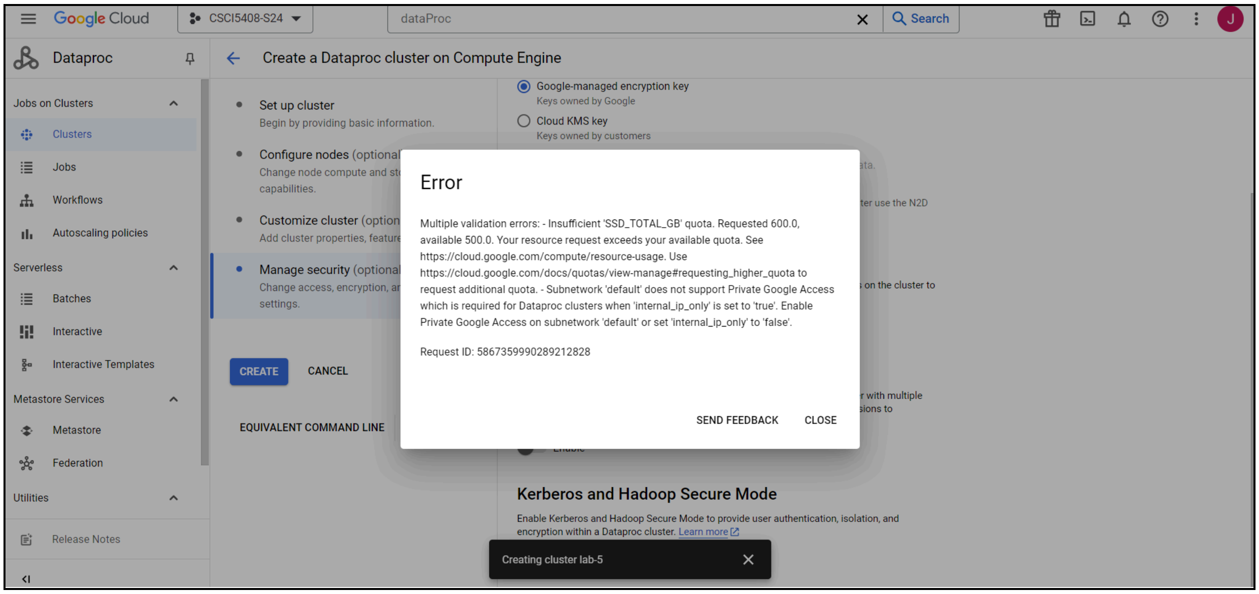
Primary disk size *: 50 GB

Primary disk type *: Balanced Persistent Disk

Number of local SSDs *: 0 x 375GB

Local SSD Interface: SCSI

Figure 6: Change machine type to n2-standard-2 for worker node.



● Google-managed encryption key
Keys owned by Google

○ Cloud KMS key
Keys owned by customers

Error

Multiple validation errors: - Insufficient 'SSD_TOTAL_GB' quota. Requested 600.0, available 500.0. Your resource request exceeds your available quota. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quotas/view-manage#requesting_higher_quota to request additional quota. - Subnetwork 'default' does not support Private Google Access which is required for Dataproc clusters when 'internal_ip_only' is set to 'true'. Enable Private Google Access on subnetwork 'default' or set 'internal_ip_only' to 'false'.

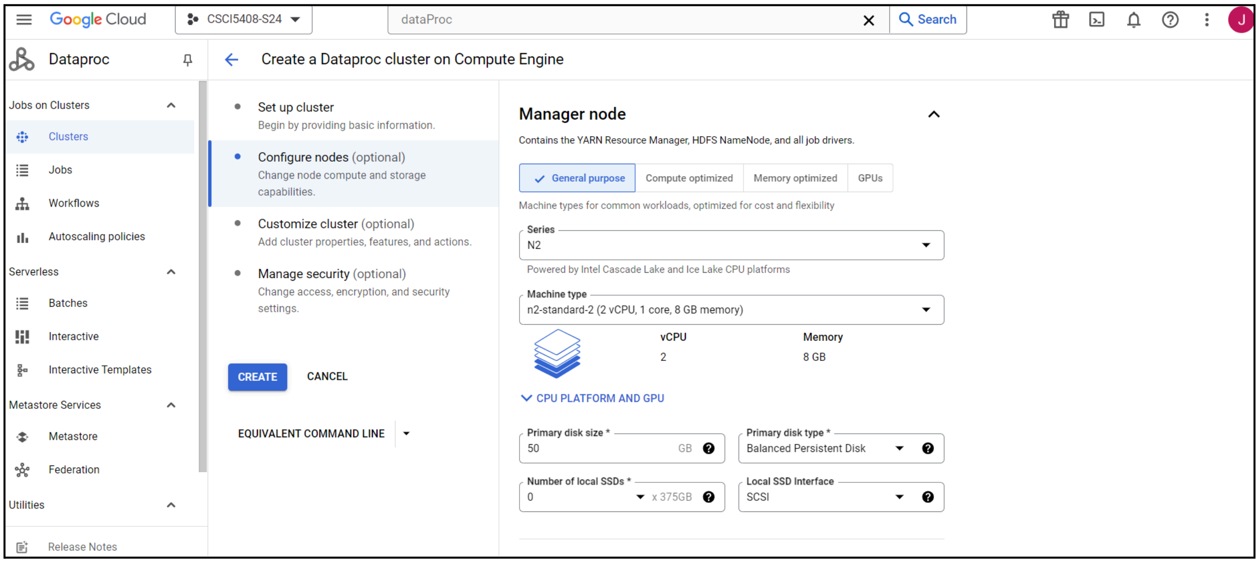
Request ID: 5867359990289212828

Kerberos and Hadoop Secure Mode

Enable Kerberos and Hadoop Secure Mode to provide user authentication, isolation, and encryption within a Dataproc cluster. Learn more

Creating cluster lab-5

Figure 7: Error due to incorrect use of machine type, and primary disk size for manager node.



Manager node

Contains the YARN Resource Manager, HDFS NameNode, and all job drivers.

✓ General purpose

Machine types for common workloads, optimized for cost and flexibility

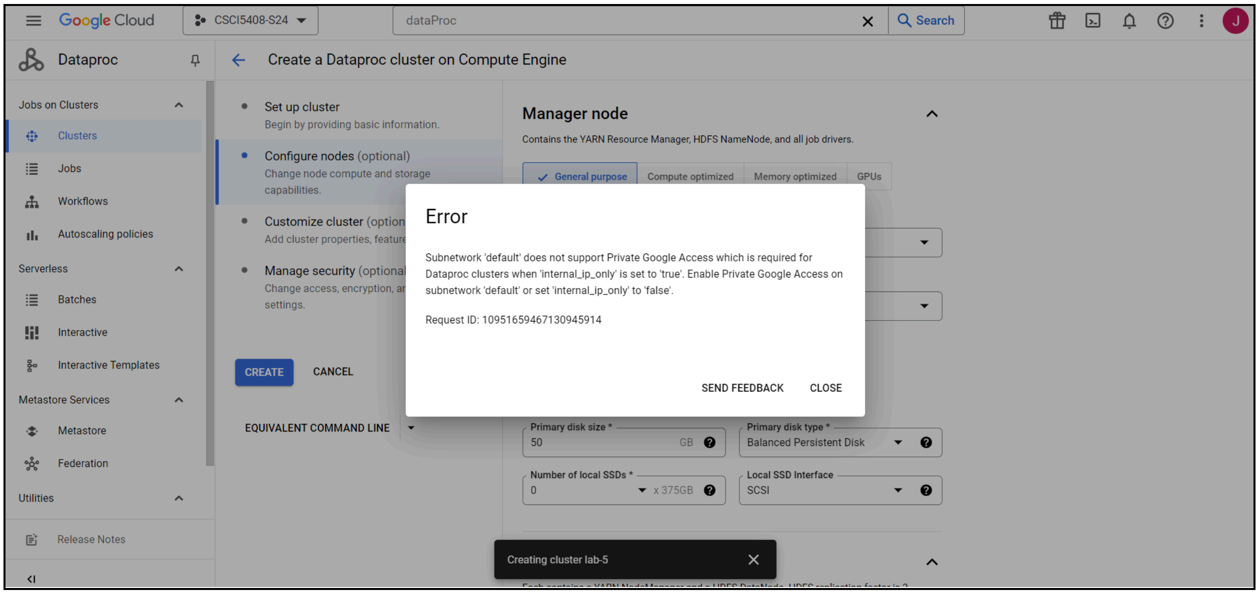
Series: N2
Powered by Intel Cascade Lake and Ice Lake CPU platforms

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

vCPU: 2
Memory: 8 GB

Primary disk size *: 50 GB
Primary disk type *: Balanced Persistent Disk
Number of local SSDs *: 0 x 375GB
Local SSD Interface: SCSI

Figure 8: Change machine type to n2-standard-2, and set disk size to 50 GB for the manager node.



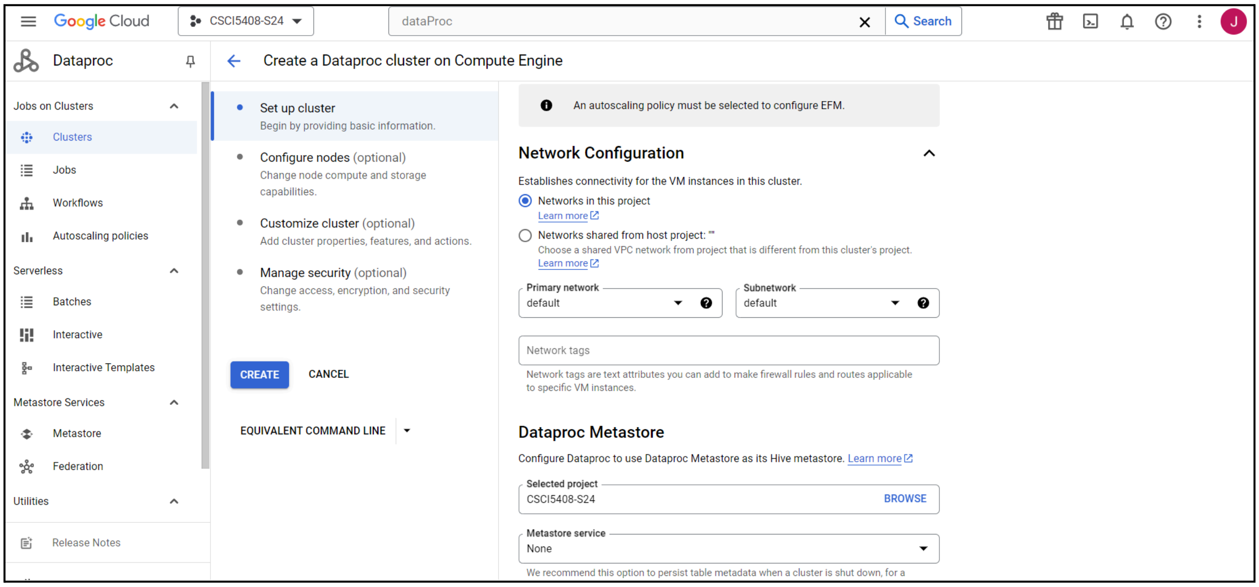
Error

Subnetwork 'default' does not support Private Google Access which is required for Dataproc clusters when 'internal_ip_only' is set to 'true'. Enable Private Google Access on subnetwork 'default' or set 'internal_ip_only' to 'false'.

Request ID: 10951659467130945914

Primary disk size *: 50 GB
Primary disk type *: Balanced Persistent Disk
Number of local SSDs *: 0 x 375GB
Local SSD Interface: SCSI

Figure 9: Error due to incorrect network configuration.


Google Cloud
CSCI5408-S24
dataProc
Search
Dataproc
Create a Dataproc cluster on Compute Engine
Jobs on Clusters
Clusters
Jobs
Workflows
Autoscaling policies
Serverless
Batches
Interactive
Interactive Templates
Metastore Services
Metastore
Federation
Utilities
Release Notes
Set up cluster
Begin by providing basic information.
Configure nodes (optional)
Change node compute and storage capabilities.
Customize cluster (optional)
Add cluster properties, features, and actions.
Manage security (optional)
Change access, encryption, and security settings.
CREATE
CANCEL
EQUIVALENT COMMAND LINE
Learn more
BROWSE

An autoscaling policy must be selected to configure EFM.

Network Configuration

Establishes connectivity for the VM instances in this cluster.

● Networks in this project

○ Networks shared from host project: ""
Choose a shared VPC network from project that is different from this cluster's project.

Primary network: default
Subnetwork: default

Network tags
Network tags are text attributes you can add to make firewall rules and routes applicable to specific VM instances.

Dataproc Metastore

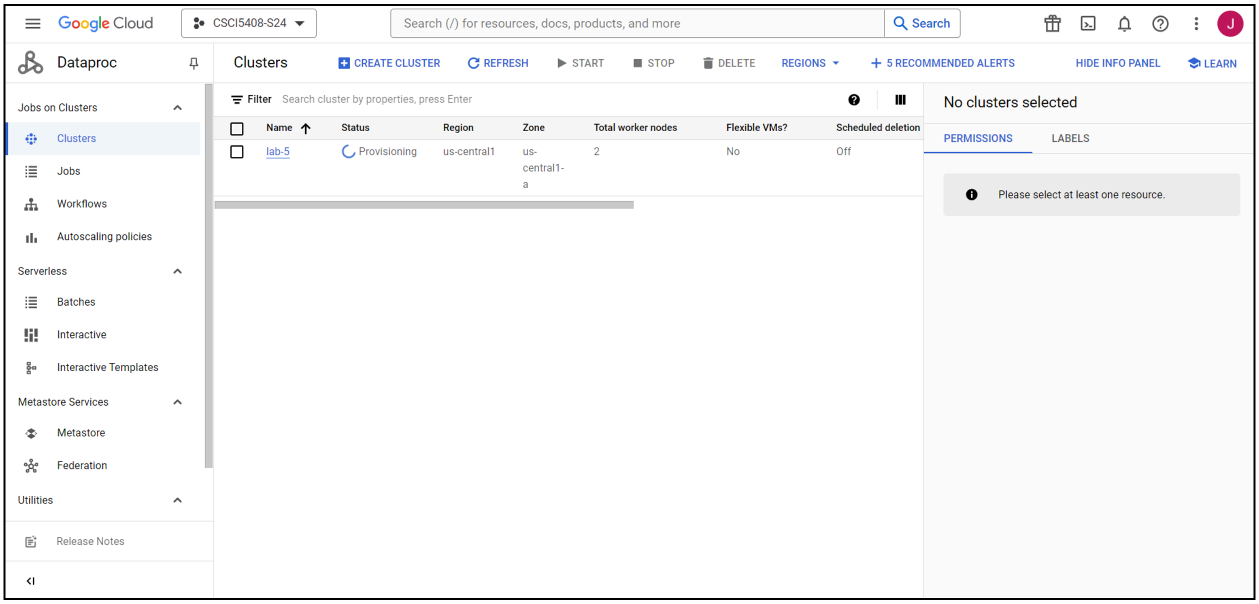
Configure Dataproc to use Dataproc Metastore as its Hive metastore.

Selected project: CSCI5408-S24

Metastore service: None

We recommend this option to persist table metadata when a cluster is shut down, for a

Figure 10: Set primary network, and sub-network to default.

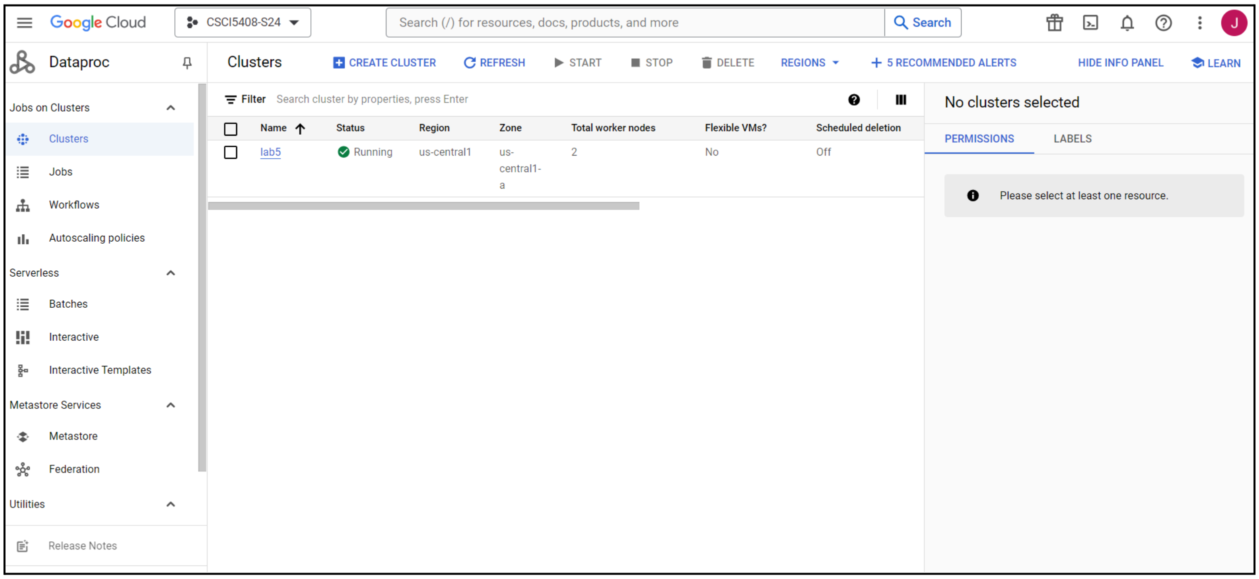


□	Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion
□	lab-5	Provisioning	us-central1	us-central1-a	2	No	Off

No clusters selected

Please select at least one resource.

Figure 11: Cluster Creation in Progress.



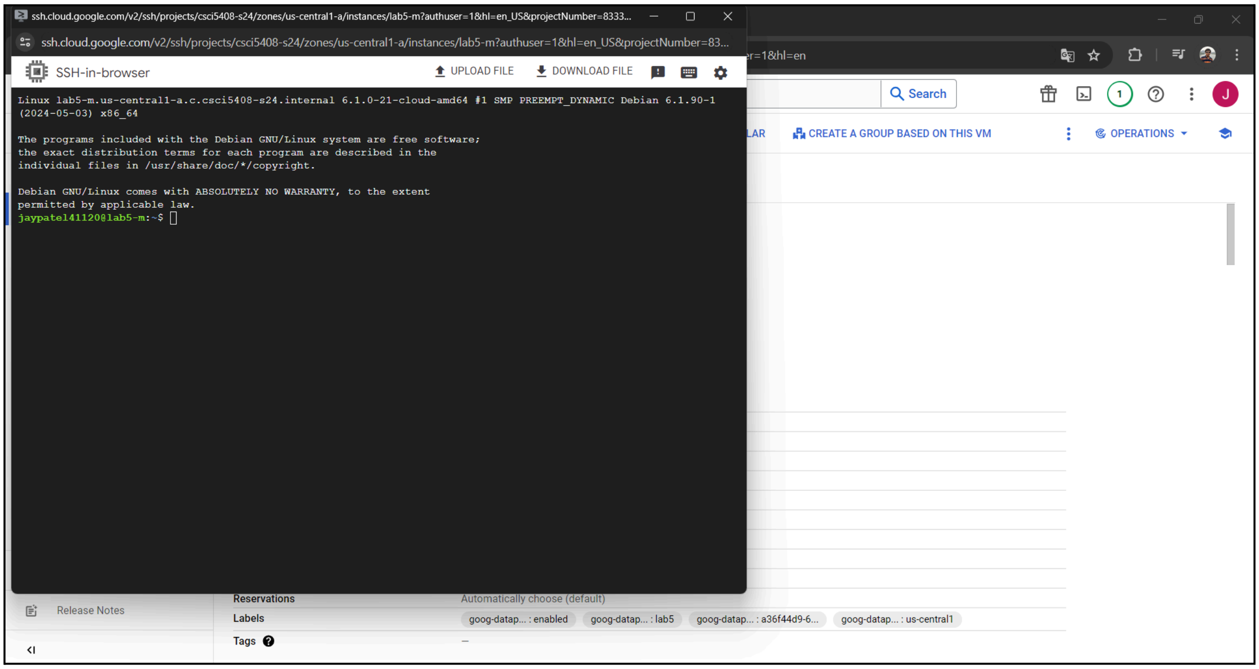
□	Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion
□	lab5	✓ Running	us-central1	us-central1-a	2	No	Off

No clusters selected

Please select at least one resource.

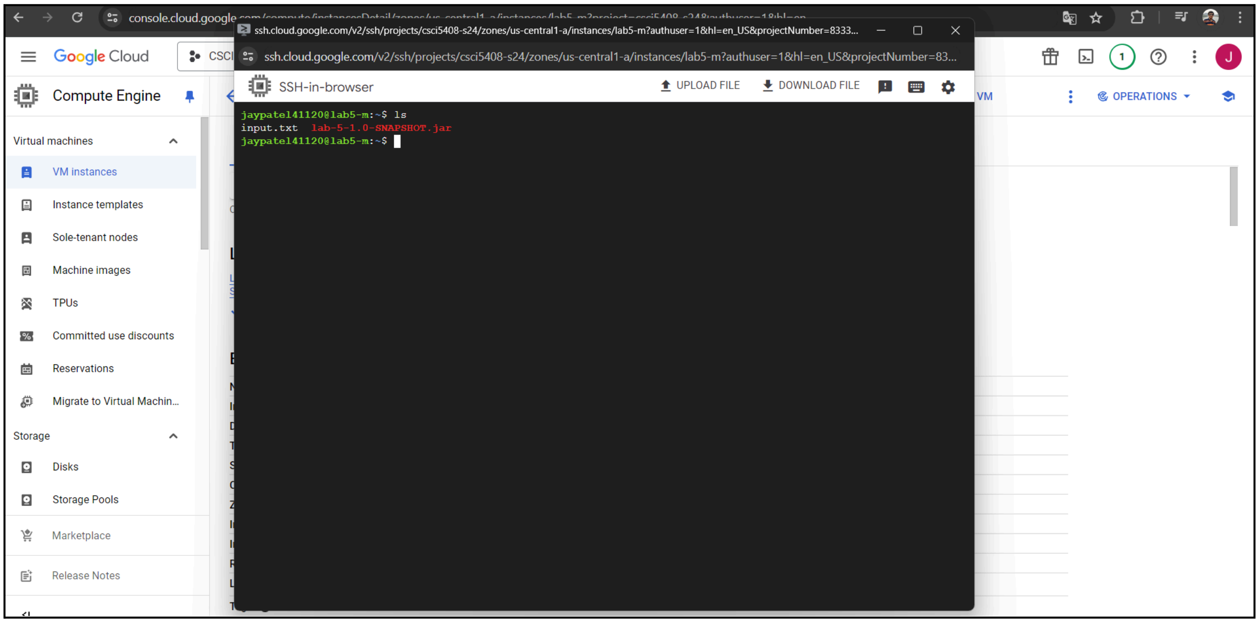
Figure 12: Apache Spark cluster is successfully created and running.

2: Problems faced while running the .jar file on a spark cluster


ssh.cloud.google.com/v2/ssh/projects/csci5408-s24/zones/us-central1-a/instances/lab5-m?authuser=1&hl=en_US&projectNumber=8333...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Search
CREATE A GROUP BASED ON THIS VM
OPERATIONS
Release Notes
Reservations
Automatically choose (default)
Labels
goog-datap... : enabled
goog-datap... : lab5
goog-datap... : a36f44d9-6...
goog-datap... : us-central1
Tags

```
Linux lab5-m.us-central1-a.c.csci5408-s24.internal 6.1.0-21-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1
(2024-05-03) 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.
jaypatel41120@lab5-m:~$
```

Figure 13: Enter into the master node and check the connection using ssh.

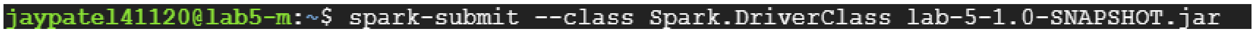


```
jaypatel41120@lab5-m:~$ ls
input.txt  lab-5-1.0-SNAPSHOT.jar
jaypatel41120@lab5-m:~$
```

Figure 14: Upload .jar and input.txt files

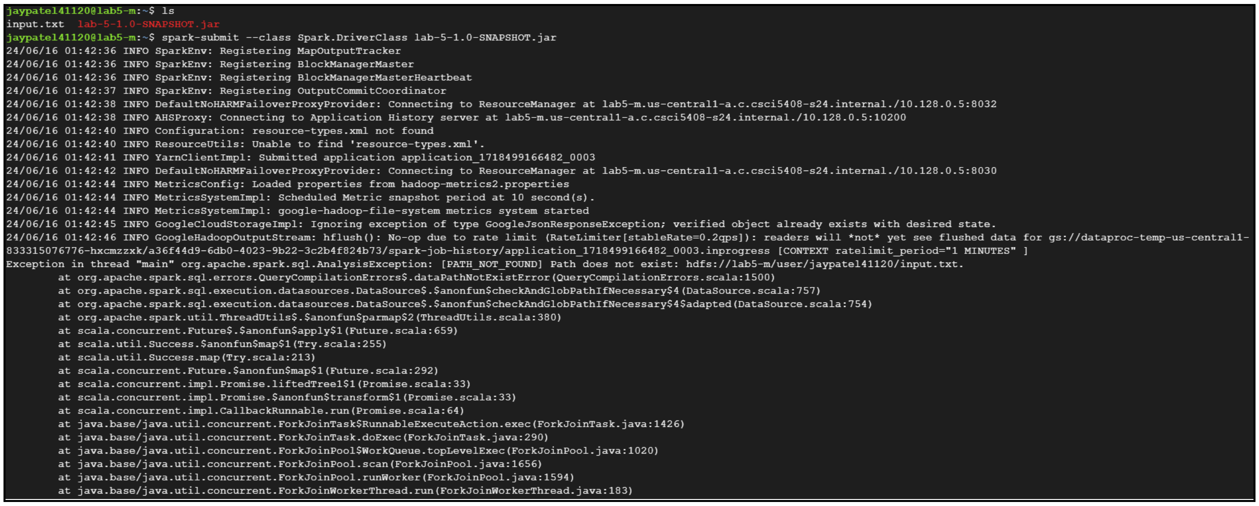
Problem 1: Unable to find the main class - cluster was not able to find the driver or main class to run the Java program

Solution: Specify that main classpath with the package name using the following command.



```
jaypatel41120@lab5-m:~$ spark-submit --class Spark.DriverClass lab-5-1.0-SNAPSHOT.jar
```

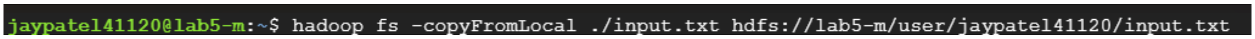
Problem 2: Path not found - program didn't find the input.txt file.



```
jaypatel41120@lab5-m:~$ ls
input.txt  lab-5-1.0-SNAPSHOT.jar
jaypatel41120@lab5-m:~$ spark-submit --class Spark.DriverClass lab-5-1.0-SNAPSHOT.jar
24/06/16 01:42:36 INFO SparkEnv: Registering MapOutputTracker
24/06/16 01:42:36 INFO SparkEnv: Registering BlockManagerMaster
24/06/16 01:42:36 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/06/16 01:42:37 INFO SparkEnv: Registering OutputCommitCoordinator
24/06/16 01:42:38 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:8032
24/06/16 01:42:38 INFO AHSProxy: Connecting to Application History server at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:10200
24/06/16 01:42:40 INFO Configuration: resource-types.xml not found
24/06/16 01:42:40 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/06/16 01:42:41 INFO YarnClientImpl: Submitted application application_1718499166482_0003
24/06/16 01:42:42 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:8030
24/06/16 01:42:44 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties
24/06/16 01:42:44 INFO MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
24/06/16 01:42:44 INFO MetricsSystemImpl: google-hadoop-file-system metrics system started
24/06/16 01:42:45 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
24/06/16 01:42:46 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushed data for gs://dataproc-temp-us-central1-
833315076776-hxcmzzxk/a36f44d9-6db0-4023-9b22-3c2b4f824b73/spark-job-history/application_1718499166482_0003.inprogress [CONTEXT ratelimit_period="1 MINUTES" ]
Exception in thread "main" org.apache.spark.sql.AnalysisException: [PATH_NOT_FOUND] Path does not exist: hdfs://lab5-m/user/jaypatel41120/input.txt.
        at org.apache.spark.sql.errors.QueryCompilationErrors$.dataPathNotExistError(QueryCompilationErrors.scala:1500)
        at org.apache.spark.sql.execution.datasources.DataSource$.$anonfun$checkAndGlobPathIfNecessary$4(DataSource.scala:757)
        at org.apache.spark.sql.execution.datasources.DataSource$.$anonfun$checkAndGlobPathIfNecessary$4$adapted(DataSource.scala:754)
        at org.apache.spark.util.ThreadUtils$.$anonfun$parmap$2(ThreadUtils.scala:380)
        at scala.concurrent.Future$.$anonfun$apply$1(Future.scala:659)
        at scala.util.Success.$anonfun$map$1(Try.scala:255)
        at scala.util.Success.map(Try.scala:213)
        at scala.concurrent.Future.$anonfun$map$1(Future.scala:292)
        at scala.concurrent.impl.Promise.liftedTree1$1(Promise.scala:33)
        at scala.concurrent.impl.Promise.$anonfun$transform$1(Promise.scala:33)
        at scala.concurrent.impl.CallbackRunnable.run(Promise.scala:64)
        at java.base/java.util.concurrent.ForkJoinTask$RunnableExecuteAction.exec(ForkJoinTask.java:1426)
        at java.base/java.util.concurrent.ForkJoinTask.doExec(ForkJoinTask.java:290)
        at java.base/java.util.concurrent.ForkJoinPool$WorkQueue.topLevelExec(ForkJoinPool.java:1020)
        at java.base/java.util.concurrent.ForkJoinPool.scan(ForkJoinPool.java:1656)
        at java.base/java.util.concurrent.ForkJoinPool.runWorker(ForkJoinPool.java:1594)
        at java.base/java.util.concurrent.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:183)
```

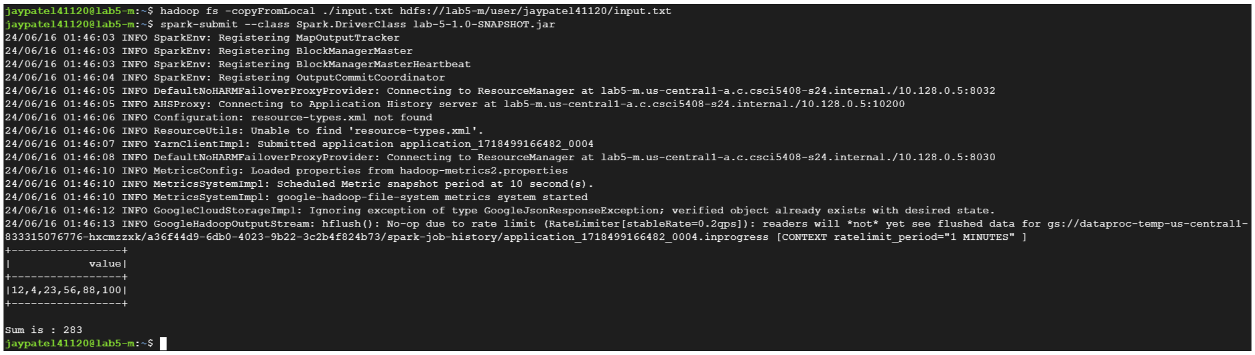
Figure 15: Error input.txt not found on the given path.

Solution: Copy the input.txt file to the Hadoop file system using the following command.



```
jaypatel41120@lab5-m:~$ hadoop fs -copyFromLocal ./input.txt hdfs://lab5-m/user/jaypatel41120/input.txt
```

Final Output

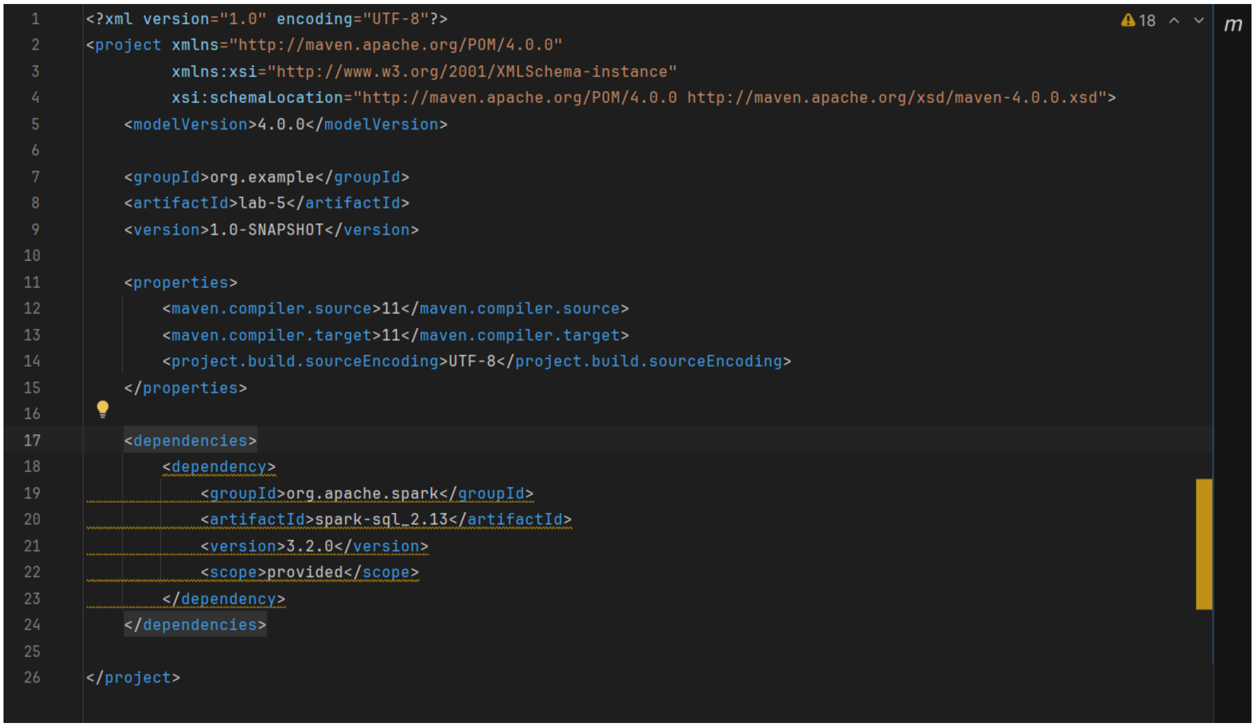


```
jaypatel41120@lab5-m:~$ hadoop fs -copyFromLocal ./input.txt hdfs://lab5-m/user/jaypatel41120/input.txt
jaypatel41120@lab5-m:~$ spark-submit --class Spark.DriverClass lab-5-1.0-SNAPSHOT.jar
24/06/16 01:46:03 INFO SparkEnv: Registering MapOutputTracker
24/06/16 01:46:03 INFO SparkEnv: Registering BlockManagerMaster
24/06/16 01:46:03 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/06/16 01:46:04 INFO SparkEnv: Registering OutputCommitCoordinator
24/06/16 01:46:05 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:8032
24/06/16 01:46:05 INFO AHSProxy: Connecting to Application History server at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:10200
24/06/16 01:46:06 INFO Configuration: resource-types.xml not found
24/06/16 01:46:06 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/06/16 01:46:07 INFO YarnClientImpl: Submitted application application_1718499166482_0004
24/06/16 01:46:08 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at lab5-m.us-central1-a.c.csci5408-s24.internal./10.128.0.5:8030
24/06/16 01:46:10 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties
24/06/16 01:46:10 INFO MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
24/06/16 01:46:10 INFO MetricsSystemImpl: google-hadoop-file-system metrics system started
24/06/16 01:46:12 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
24/06/16 01:46:13 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushed data for gs://dataproc-temp-us-central1-
83331507676-hxcmzzxk/a36f44d9-6db0-4023-9b22-3c2b4f824b73/spark-job-history/application_1718499166482_0004.inprogress [CONTEXT ratelimit_period="1 MINUTES" ]
+-----------------+
|            value|
+-----------------+
|12,4,23,56,88,100|
+-----------------+

Sum is : 283
jaypatel41120@lab5-m:~$ 
```

Figure 16: Final Output

2: Java Code Explanation

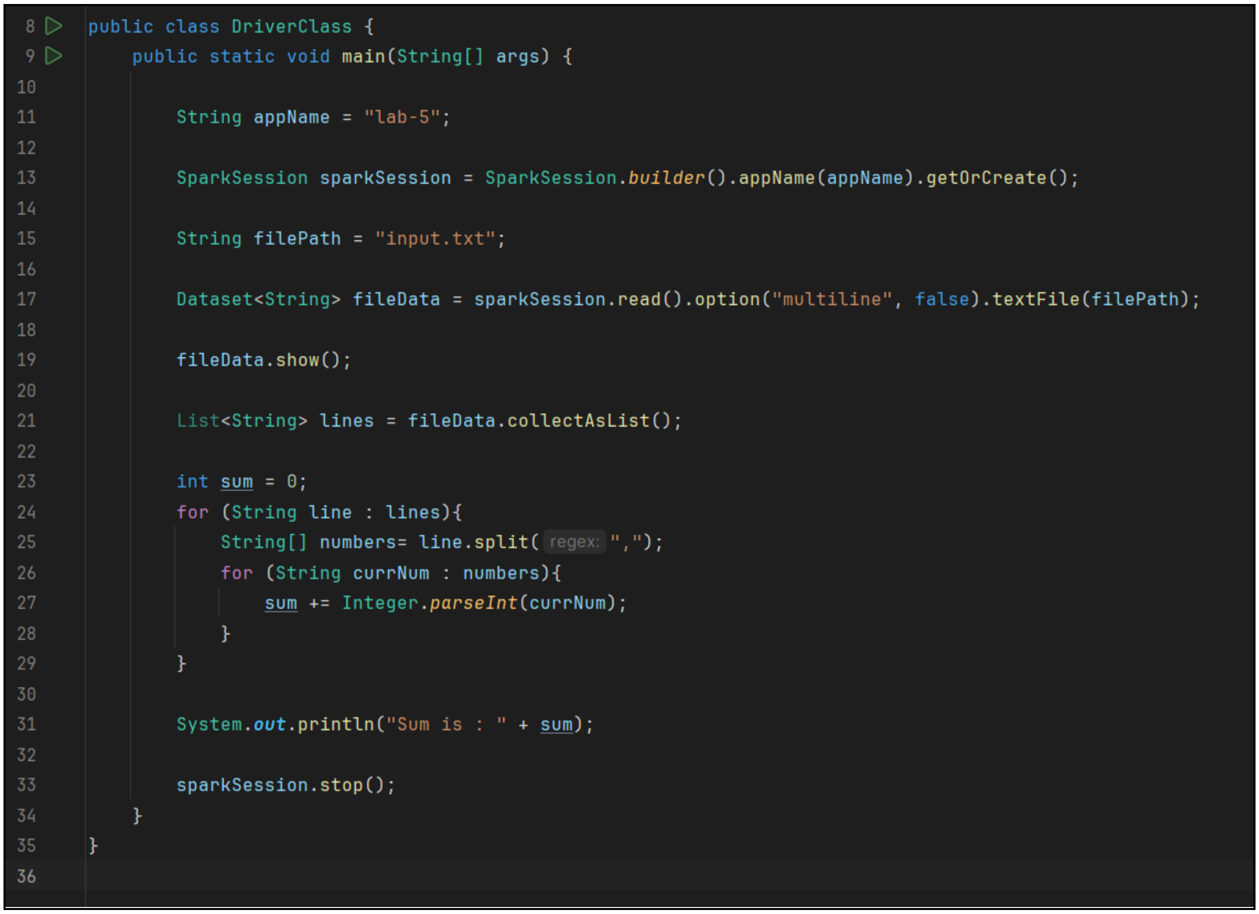
pom.xml (Insert Apache Spark dependency)



```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>lab-5</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-sql_2.13</artifactId>
            <version>3.2.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

</project>
```

Figure 17: pom.xml file of Java program

- Here I use Apache Spark version 3.2.0 and I also have to change the compiler version from 21 to 11 because when I initially run using the 21 version it gives me an error on the GCP cluster, so after that, I just change to 11 and it works for me.



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

        String appName = "lab-5";

        SparkSession sparkSession = SparkSession.builder().appName(appName).getOrCreate();

        String filePath = "input.txt";

        Dataset<String> fileData = sparkSession.read().option("multiline", false).textFile(filePath);

        fileData.show();

        List<String> lines = fileData.collectAsList();

        int sum = 0;
        for (String line : lines){
            String[] numbers= line.split( regex: ",");
            for (String currNum : numbers){
                sum += Integer.parseInt(currNum);
            }
        }

        System.out.println("Sum is : " + sum);

        sparkSession.stop();
    }
}
```

Figure 18: Java Main Class

- Here I first start the spark session by providing any random app name, and after that set the file path to read the input.txt file.
- Show the contents of the file to the console.
- Convert the file data into a list of strings.
- Now here we have a list so we have to iterate through each object of that list and here we have only one object as we have only one line in our input.txt file.
- Now in the loop first we have to split the line using the "," operator and we get the list in which all the numbers are in string format.
- Now just change numbers from string to integer and add them into the global variable named sum.
- Show the sum of all the numbers to the console.
- Stop the spark session.