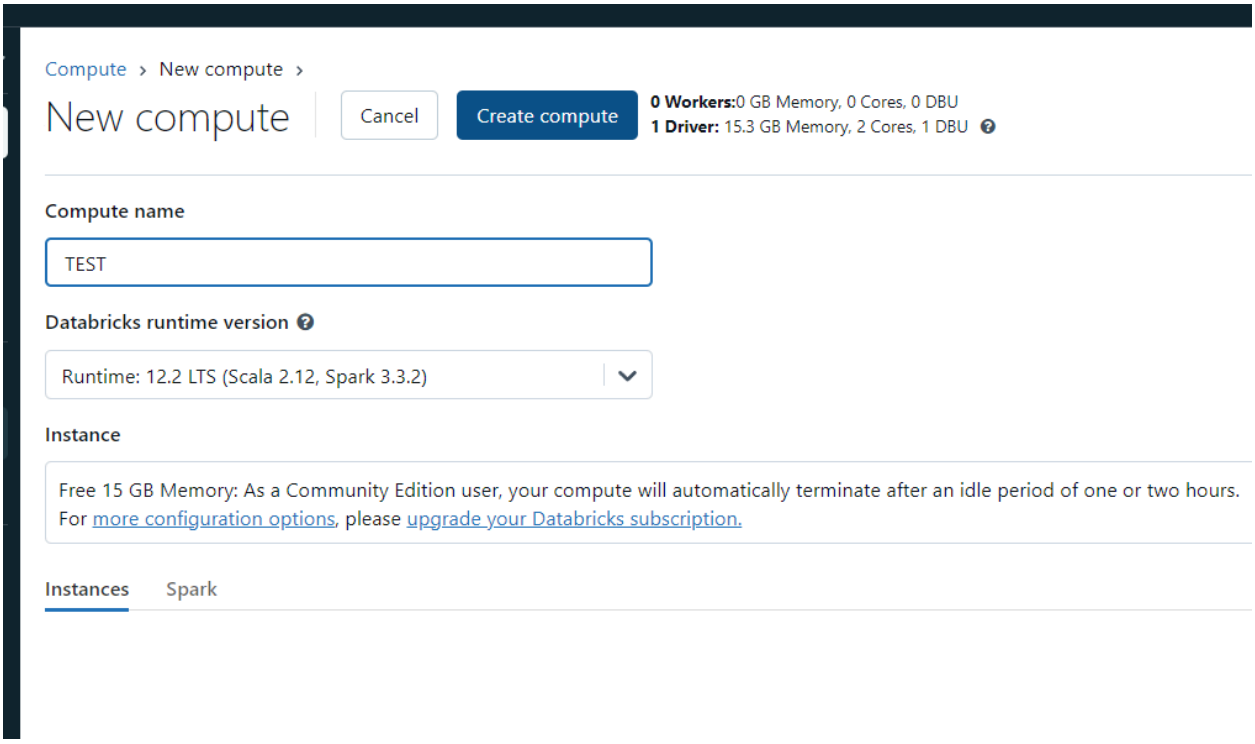
DAY-12 (Spark Assignment - 2)
Mitushi Vishwakarma

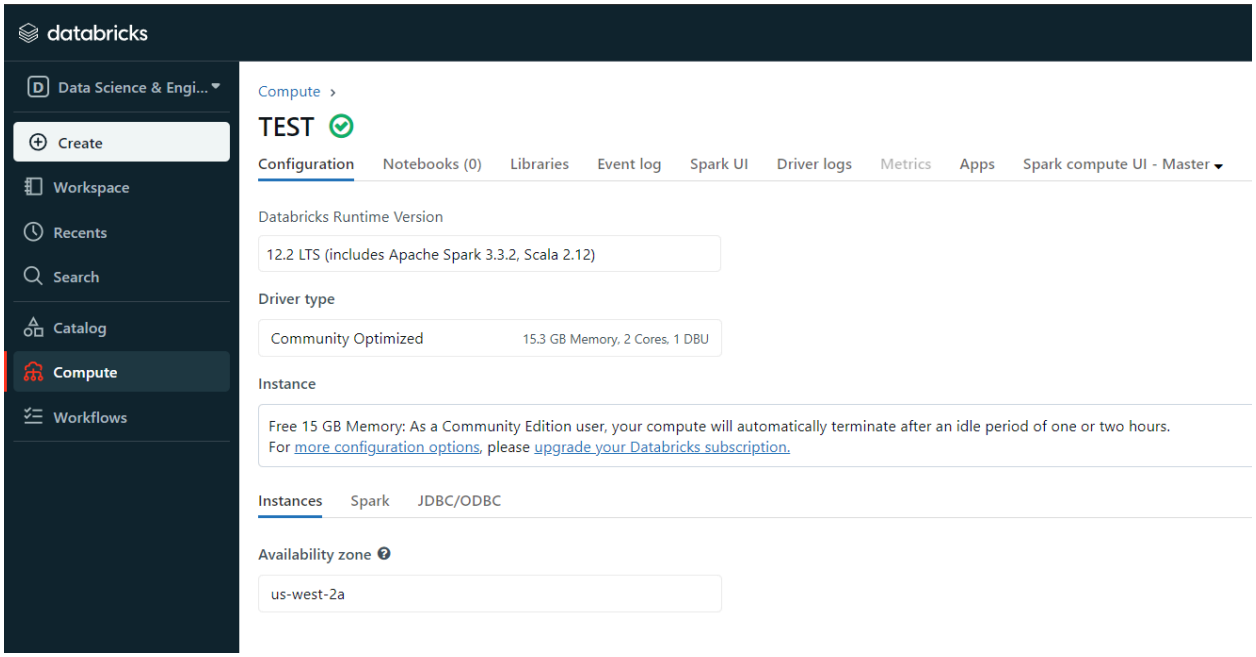
Creating Compute in DataBricks :

Databricks is a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale.

Databricks compute refers to the selection of computing resources you can provision in your Databricks workspace. We need compute to run data engineering, data science, and data analytics workloads



Compute created successfully.



Before using pyspark in jupyter notebook, We need to run the following command
Pip install pyspark
Then from pyspark.sql module import SparkSession method to create a session as a variable. Then using spark variable we read csv file using read.csv() and displayed data using .show ().

```
[2]: import pyspark

[22]: from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("sampleData").getOrCreate()
data = spark.read.csv("Employees.csv")
data.show()

+-----+-------+-------+
|  _c0|    _c1|    _c2|
+-----+-------+-------+
|EmpID|   Name| Salary|
| e101| Pramod|1200000|
| e120| Dinesh|2200000|
| e205|Sabesta|1500000|
| e331|  Harry|1700000|
| e421|Avinash|1300000|
| e231|    Joy|2300000|
| e222|  Smith|2100000|
| e339|   Khan|1800000|
| e150|  Dilip|1900000|
| e131|  Kiran| 800000|
+-----+-------+-------+

[23]: data

[23]: DataFrame[_c0: string, _c1: string, _c2: string]
```

Here we created spark session then using in code collection of objects and passing that collection into SparkContext.parallelize and using .collect() to display data.

First_notebook Python

File Edit View Run Help Last edit was 4 hours ago New cell UI: OFF

Cmd 1

```python
from pyspark.sql import SparkSession

spark = SparkSession.builder \
      .master("local[1]") \
      .appName("SparkByExamples.com") \
      .getOrCreate()
dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]
rdd=spark.sparkContext.parallelize(dataList)
rdd.collect()
```

(1) Spark Jobs

```
Out[1]: [('Java', 20000), ('Python', 100000), ('Scala', 3000)]
```

Command took 0.58 seconds -- by mitushivishrgpv@gmail.com at 2/5/2024, 3:15:14 PM on TEST

Reading csv file using textFile() method :

Cmd 2

```python
rdd2 = spark.sparkContext.textFile("/FileStore/tables/employee/Employees.csv")
rdd2.collect()
```

(1) Spark Jobs

```
Out[2]: ['EmpID,Name,Salary',
 'e101,Pramod,1200000',
 'e120,Dinesh,2200000',
 'e205,Sabesta,1500000',
 'e331,Harry,1700000',
 'e421,Avinash,1300000',
 'e231,Joy,2300000',
 'e222,Smith,2100000',
 'e339,Khan,1800000',
 'e150,Dilip,1900000',
 'e131,Kiran,800000']
```

Command took 1.68 seconds -- by mitushivishrgpv@gmail.com at 2/5/2024, 3:21:03 PM on TEST

Reading text file using textFile() method in pyspark.

```
rdd3 = spark.sparkContext.textFile("/FileStore/tables/employee/Testfile-1.txt")
rdd3.count()
```

(2) Spark Jobs

```
Out[1]: 2
```

Command took 10.46 seconds -- by mitushivishrgpv@gmail.com at 2/5/2024, 9:58:22 PM on TEST

Cmd 4

```
rdd3.collect()
```

(1) Spark Jobs

```
Out[2]: ['Hello Everyone! ', 'Wwlcome to pyspark programming']
```

Command took 0.89 seconds -- by mitushivishrgpv@gmail.com at 2/5/2024, 9:59:32 PM on TEST

Day - 2

Agenda :-

1. Intro. to PySpark and setting up env.
2. Hands-on → Basic Pyspark
3. —"— → analyze Sample dataset
4. RDDs and transformations
5. Hands-on → Transforming data with Pyspark RDDs.

Pyspark → Apache Spark library ~~in~~ written in python.
to run Python applications using Apache Spark capabilities.

→ Python API
→ analytical processing engine for large scale data processing and ML applications

Spark → data analytics application
can run on → Single node machines → one worker
multi-node machines → multiple workers.

→ created to ~~reduce~~ ~~lim~~ address the limitations of mapReduce.
→ in-memory processing.

Pyspark :-

→ efficient processing of large datasets

→ can be used in Anaconda, Spyder, Jupytes.

Features -

→ in - memory computation.

→ parallel processing

→ used with yarn, Mesos

→ fault tolerant -

→ immutable

→ Lazy evaluation

→ cache & persistence

→ inbuild optimization

App. run on Pyspark is 100x faster than traditional system.

→ Using pyspark, we can process data from Hadoop. HDFS, AWS S3.

→ used to process real-time data using Kafka

Apache Kafka.

→ event streaming platform

→ high-performance data pipelines.

Pyspark supports :-
Python 3.8
Java 8, 11, 13, 14
Scala 2.12 and 2.13
R 3.5

Modules & Packages

PySpark, RDD
pyspark.sql (Dataframe and sql)
pyspark.streaming
pyspark.ml / mllib
pyspark.GraphFrames
pyspark.resource.

Pyspark commands

Read data of a file.
→ create a txt file.
→ val filePath = " ___path ___
→ val textRDD = sc.textFile(filePath)
→ textRDD.collect().foreach(println)

Create RDD

2 ways
→ loading an external dataset
→ distributing a set of collection of objects

using parallelize function.

RDD is an immutable distributed collection of object sets

SUNDAY 24 → data structure in pyspark.

parallelize() takes collection of objects and pass it to Spark Context

Spark Context is the entry point to Spark cluster manager.

SC → Spark Context in spark-shell

pyspark.sql module → SparkSession method

getOrCreate() → initialize the session.

- show() will display the data
- toDF() is used to create DataFrame with specified column.

Spark.sparkContext.parallelize([ ]).toDF([columns])

Spark.stop → terminates the sparkSession.

Spark.createDataFrame (data = , Schema =)

First step :- 1. import SparkSession
2. create sparkSession.
3. add data
4. display
5. stop session.

Create RDD
1. First create SparkSession
→ using builder ()
or
→ newSession()

We can create multiple Spark Session but one sparkContext.