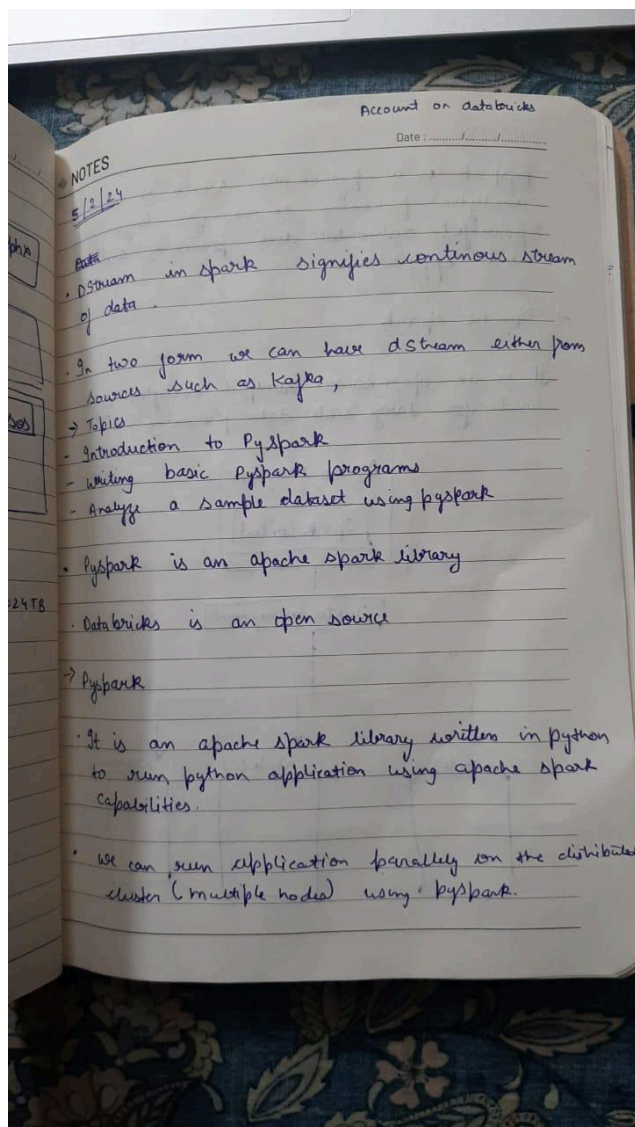


Asmita Porwal  
Batch-1  
Day-12  
5/2/2024  
Data engineering

## Assignment-12

Hand written notes during the session :

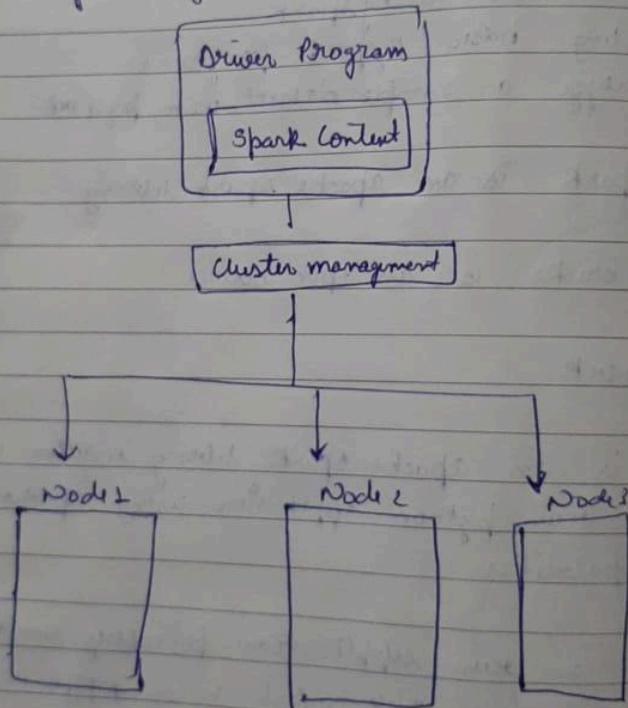


## ◆ NOTES

PySpark is a python API which is an analytical processing engine for large-scale powerful distributed data processing and ml applications.

→ Apache Spark

It is an open source unified analytics engine used for large scale data processing.



## NOTES

Date: \_\_\_\_\_

### → Use PySpark

It is used in the Data science and Machine learning community as there are many

It is used by many organization like Walmart, Sanofi

JupyterLab, Anaconda we can use

### → Features of pyspark

- In memory computation
- Distributed processing using parallelly
- Immutable
- Lazy evaluation
- Cache & persistence
- Inbuilt optimization when using Data frames
- support ANSI SQL.
- Fault tolerant
- Can be used with many cluster managers (spark, Yarn, Mesos etc)

◆ NOTES

→ Advantages of pyspark

- pyspark is a general purpose, in memory, distributed processing engine that allows you to process data efficiently in a distributed fashion.
- Applications running on pyspark are 100x faster than traditional systems.
- You will get great benefits from using pyspark for data ingestion pipeline.
- Using pyspark we can process data from Hadoop.

- Version of python pyspark

Language      Version

Python : 3.8

Java : Java 8, 11, 13, 17

Scala : 2.12 and 2.13 beyond.

R : 3.5

## NOTES

Date: .....

- Apache Kafka is an open source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration and mission-critical application

Support for Java 8 versions prior to 8U371 has been deprecated starting from Spark 3.5.0.

- Cluster Manager

- Modules & packages

- PySpark RDD (pyspark.rdd)
- PySpark Dataframe and SQL (pyspark.sql)
- " Streaming (pyspark.streaming)
- " MLlib (pyspark.ml, pyspark.mllib)
- " GraphFrames (GraphFrames)
- " Resource (pyspark.resource) It's new in PySpark 3.0.



## ◆ NOTES

- Create RDD

Two ways to create RDD.

① Loading an external data set or distributing a set of collection of objects

- `parallelize()` function which takes an already existing collection in your program and pass the same to the spark context

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession
```

```
• builder
```

```
• appName("Python Spark create RDD example")
```

```
• config("spark some config option")
```

• we need to create a spark

**Create RDD on data bricks**



Cmd 1

```

1 import pyspark
2 from pyspark.sql import SparkSession
3 spark = SparkSession.builder.appName("practice").getOrCreate()
4
5 spark

```

SparkSession - hive

SparkContext

[Spark UI](#)

Version

v3.5.0

Master

local[8]

AppName

Databricks Shell

Command took 0.98 seconds -- by asmitaporwal0404@gmail.com at 2/5/2024, 3:14:48 PM on My Cluster

Cmd 2

```

1 from pyspark.sql import SparkSession
2
3
4 # Create SparkSession
5 spark = SparkSession.builder \
6     .master("local[1]") \
7     .appName("SparkByExamples.com") \
8     .getOrCreate()
9 dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]
10 rdd=spark.sparkContext.parallelize(dataList)
11 rdd.collect()
12 result = rdd.collect()
13 ("RDD Contents:", result)

```

▶ (1) Spark Jobs

[('Java', 20000), ('Python', 100000), ('Scala', 3000)]

Command took 2.54 seconds -- by asmitaporwal0404@gmail.com at 2/5/2024, 3:14:48 PM on My Cluster

[Shift+Enter] to run

[Shift+Ctrl+Enter] to run selected text

## Creating rdd on local system on jupyter notebook

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

```
In [2]: spark = SparkSession.builder.appName("Jupyter Notebook").getOrCreate()
```

```
In [3]: spark
```

```
Out[3]: SparkSession - in-memory
SparkContext
```

[Spark UI](#)

Version

v3.5.0

Master

local[\*]

AppName

Jupyter Notebook

```
In [4]: df = spark.read.csv("E:\\downloads\\Marks_data.csv")
df
```

```
Out[4]: DataFrame[_c0: string, _c1: string, _c2: string, _c3: string]
```

```
In [5]: df.show()
```

```
+---+-----+-----+
|_c0|_c1|_c2|_c3|
+---+-----+-----+
|Name|M1 Score|M2 Score|age|
|Alex|62|80|20|
|Brad|45|56|19|
|Joey|85|98|21|
|abhi|54|79|20|
+---+-----+-----+
```

```
In [6]: dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]
```

```
In [7]: rdd = spark.sparkContext.parallelize(dataList)
```

```
In [8]: result = rdd.collect()
result
```

```
Out[8]: [('Java', 20000), ('Python', 100000), ('Scala', 3000)]
```

```
In [10]: rdd = spark.sparkContext.textFile("D:\\pyspark.txt")
```

```
In [11]: result = rdd.collect()
result
```

```
Out[11]: ['PySpark has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for S
park. In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programmi
ng language']
```