PySpark

Apache Spark is one of the open-source frameworks for distributed computing

Assume that we are supposed to find the average marks of students for each subject. So, when the job is submitted to Spark, it immediately allocates a supervisor (Driver) to over look the job submitted. This supervisor gives instructions to the Executors about the 'Task' that they are supposed to complete. 'Driver' program performs its various operation by communicating to Spark through a 'SparkContext' object. SparkContext is the starting point of any spark application. So, if we were to write a Spark Application to find the sum of marks of students, the first objective would be to create an object of SparkContext.

1. Spark Interactive Shell
2. Execute Spark Script

As mentioned previously, SparkContext is the starting point of any spark application. So, an object of spark context has to be first created in order to perform any operations in the application.

Spark context can be created using 'SparkContext' class present in 'pyspark' module.  
So, the process of writing a simple PySpark application would be

1. Import the required modules to access SparkContext class.
2. Create a SparkContext object with the required configuration.
3. Make use of the created SparkContext object to achieve the goal of the application

from pyspark import SparkContext

sc1 = SparkContext(master='local',appName='test1')

iris1 = sc1.textFile("./dataset/iris\_site.csv")

print(iris1.collect())

In the example shown above, the following two configurations are set:

* **master**- Spark Master URL to connect to
* **appName**- set application name

**Executing PySpark Scripts**

All the PySpark logic is loaded into a python script and executed. For example, here we are executing a python file - my\_application.py using the spark-submit command.

spark-submit my\_application.py

Once Spark services are started, it provides a Web-URL with which we can check its status. The URL can be retrieved using the following command.

sc.uiWebUrl