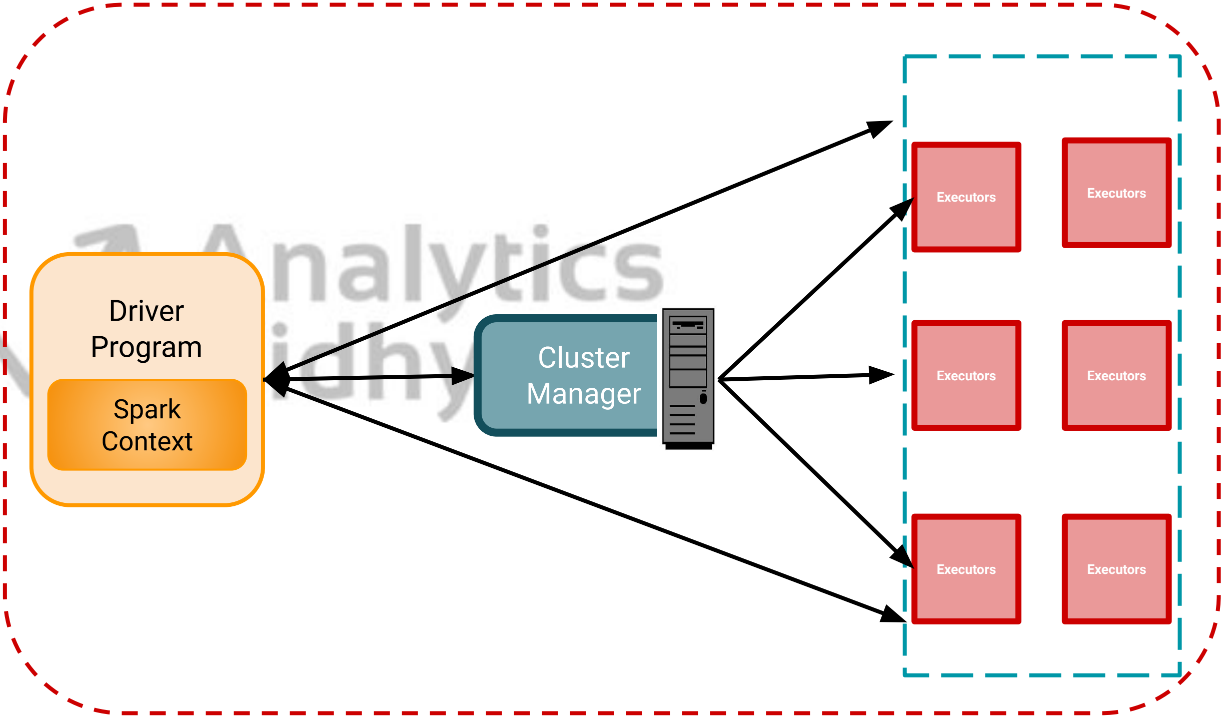
Spark Context vs Spark Session

Spark Context

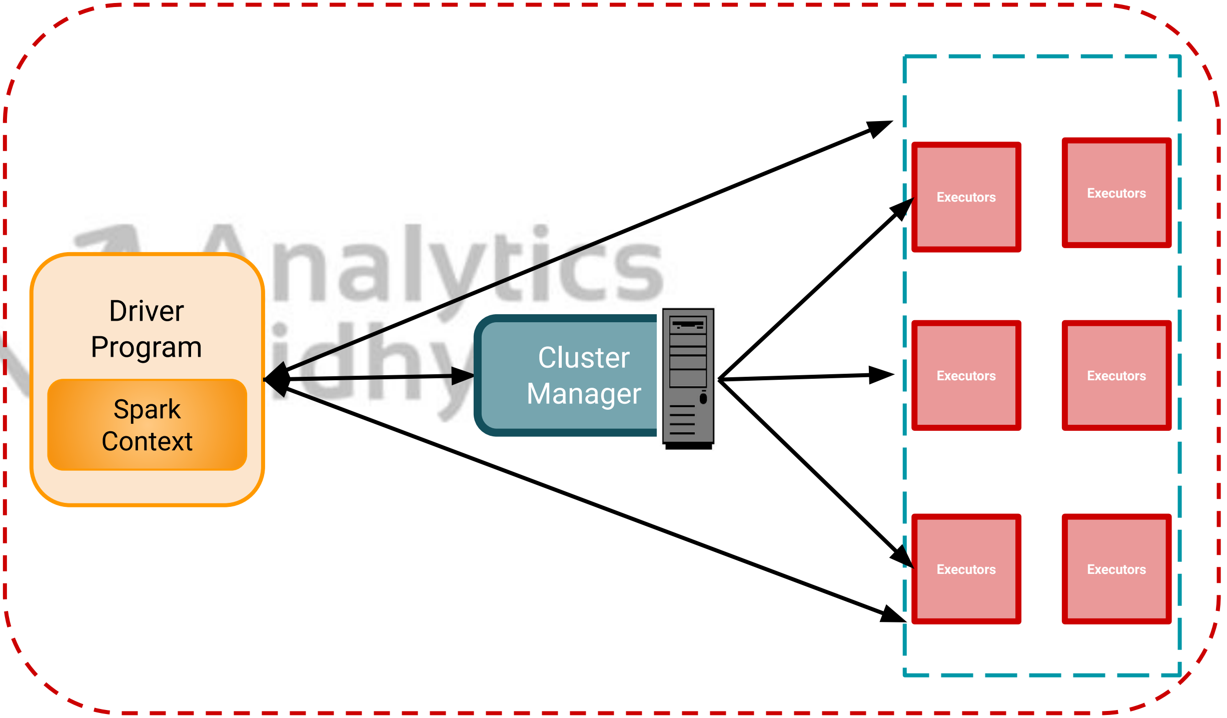
Entry point to any spark functionality.



Spark Context

Entry point to any spark functionality.

SparkContext is initiated inside Driver Program.

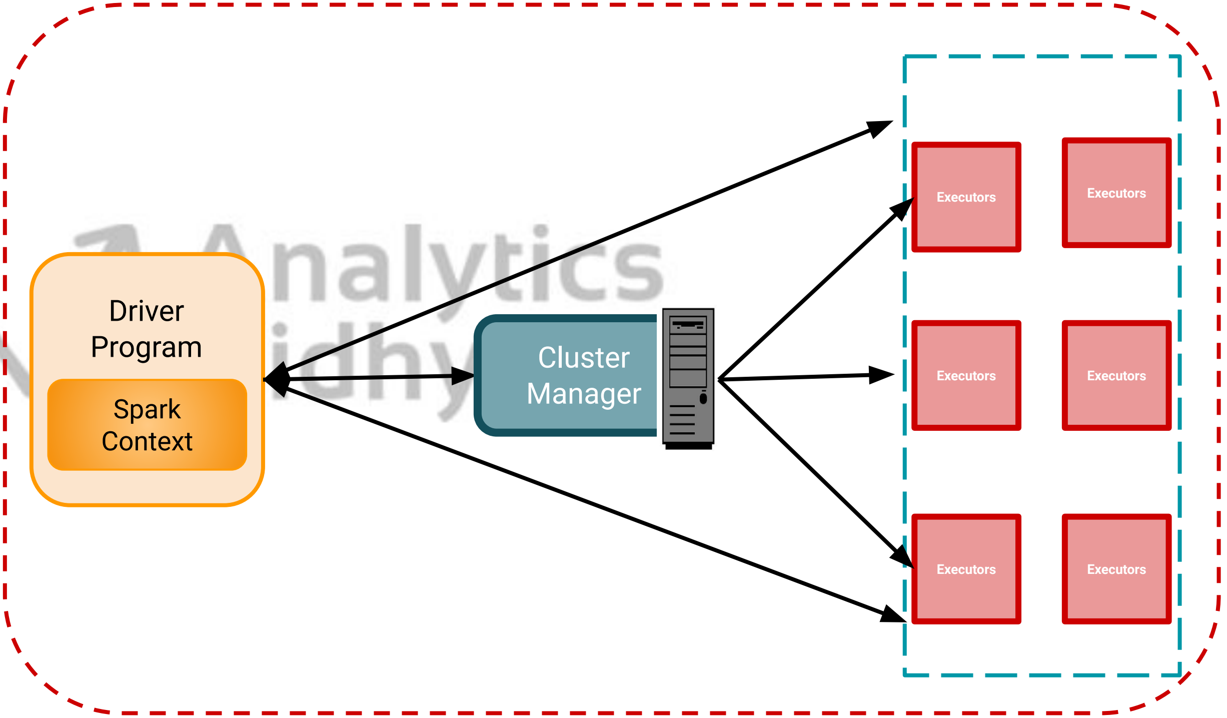


Spark Context

Entry point to any spark functionality.

SparkContext is initiated inside Driver Program.

Driver Program then runs the operations inside the Executors.



Spark Context

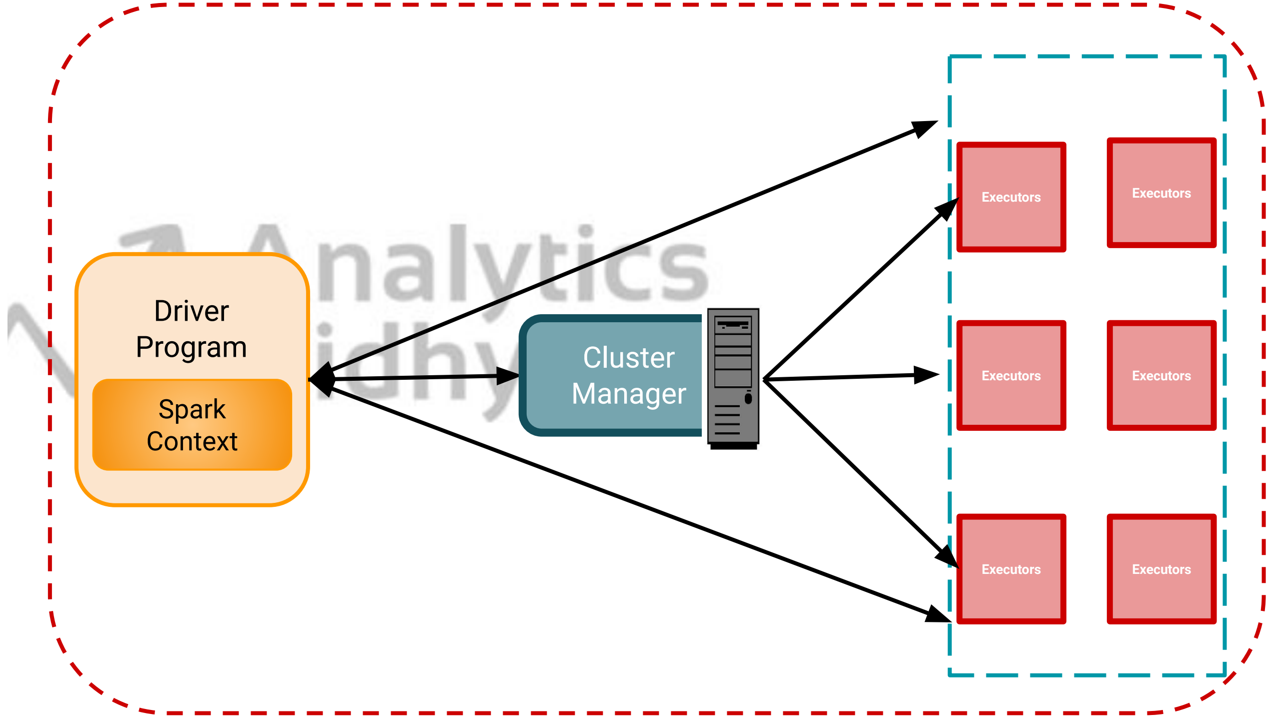
Submit jobs

Ask for resources

Schedule Tasks

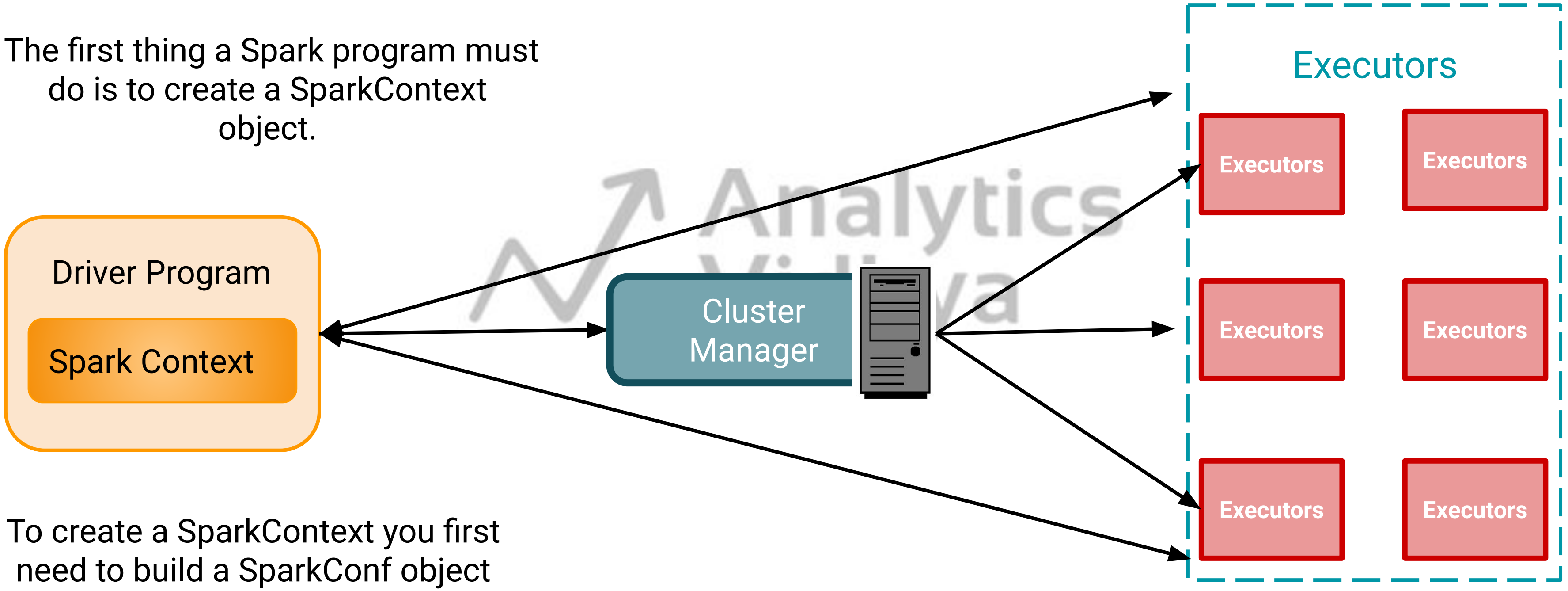
Send Tasks

Get Job results



Spark Context

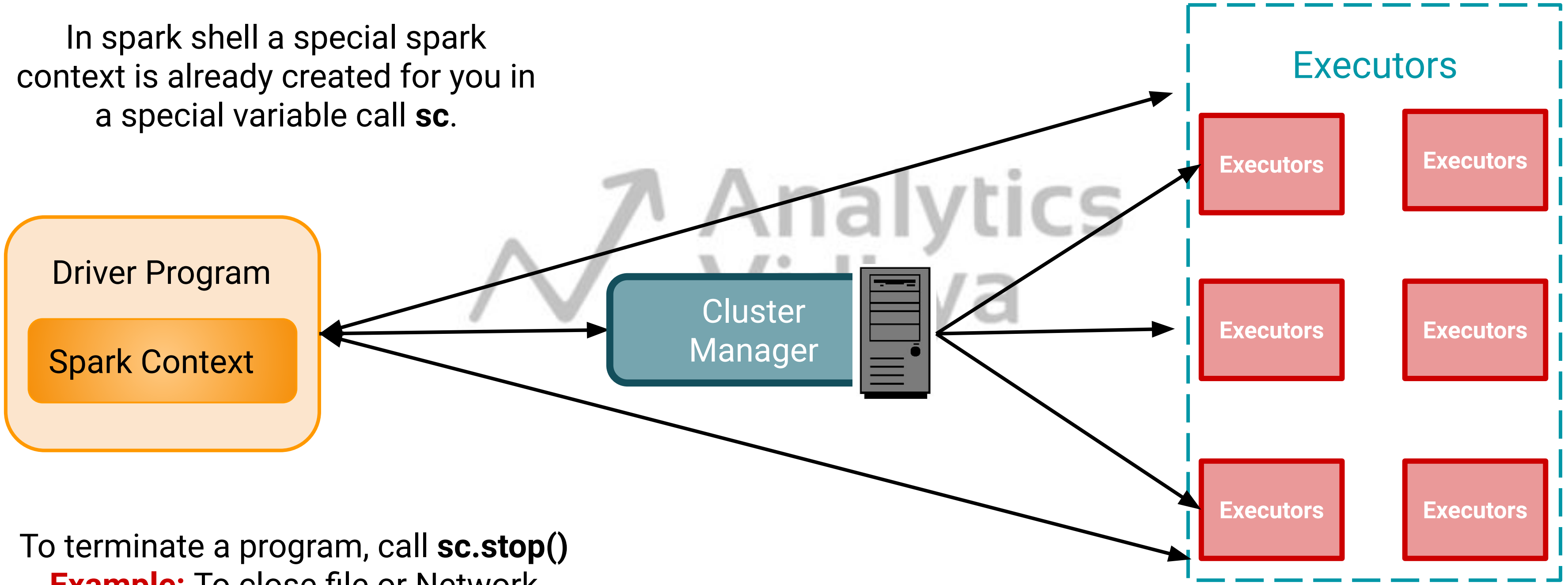
The first thing a Spark program must do is to create a SparkContext object.



To create a SparkContext you first need to build a SparkConf object that contains information about your application.

Spark Context

In spark shell a special spark context is already created for you in a special variable call **sc**.

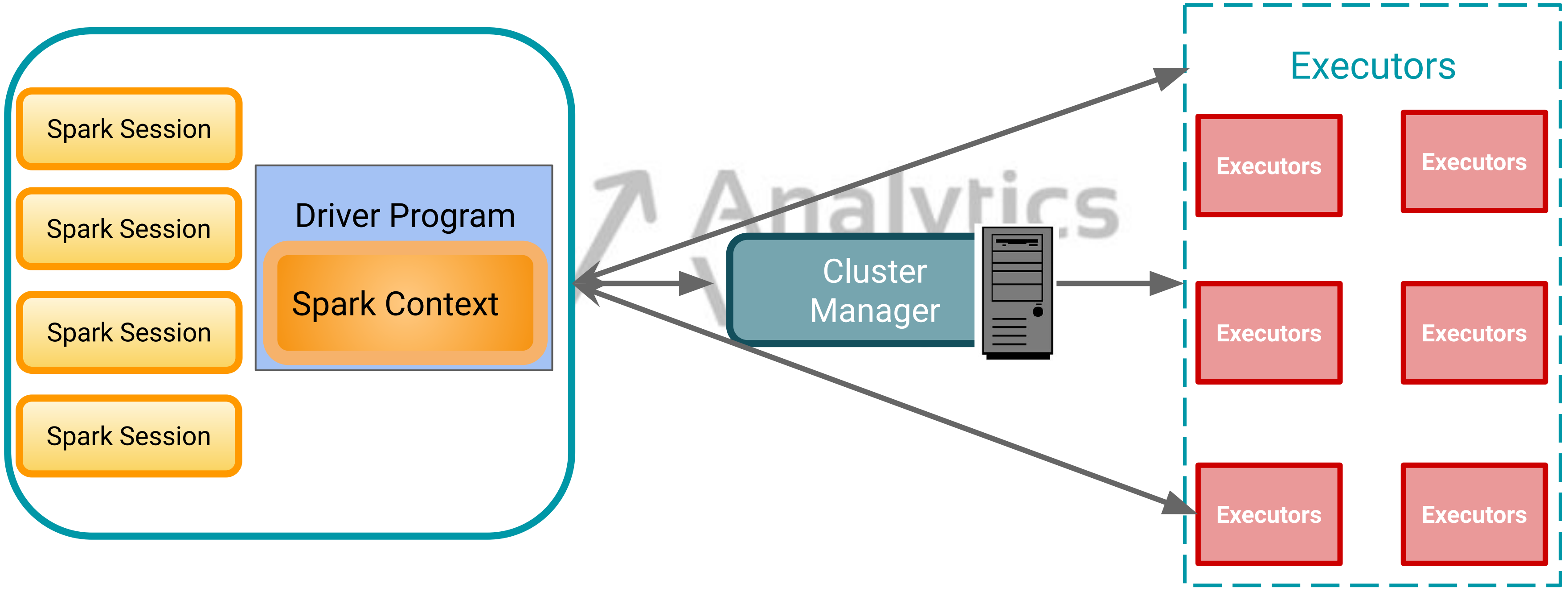


To terminate a program, call **sc.stop()**
Example: To close file or Network connections

Creating a SparkContext

```
conf = SparkConf().setAppName(appName).setMaster(master)
sc = SparkContext(conf=conf)
```

Spark Session

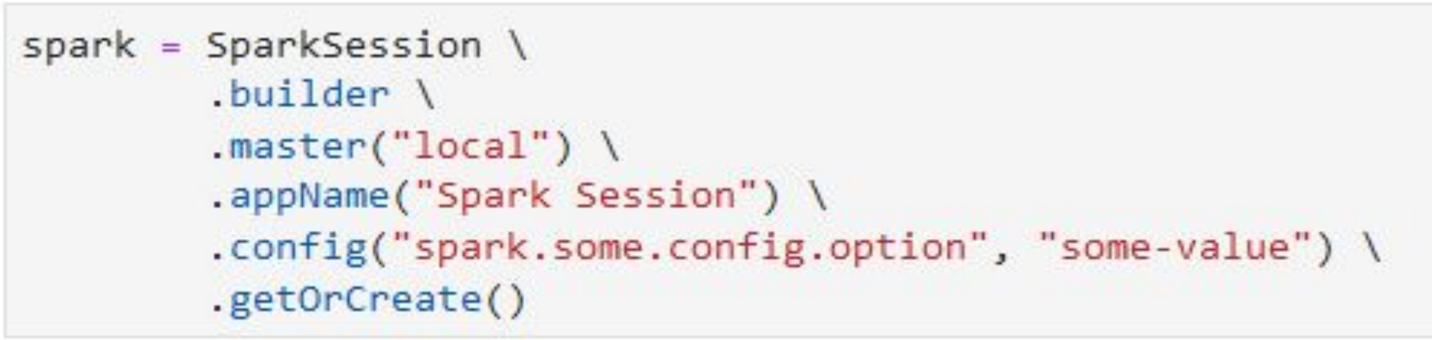


Creating a SparkSession

In previous versions of Spark, you had to create a SparkConf and SparkContext to interact with Spark, as shown previously.

In Spark 2.0 the same effects can be achieved through **SparkSession**, without explicitly creating **SparkConf**, **SparkContext** or **SQLContext**, as they're encapsulated within the SparkSession.

// Create a SparkSession. No need to create SparkContext
// You automatically get it as part of the SparkSession



```
spark = SparkSession \
        .builder \
        .master("local") \
        .appName("Spark Session") \
        .config("spark.some.config.option", "some-value") \
        .getOrCreate()
```

Thank You