**PySPARK in Windows using Jupyter Notebook**

Many tutorials are available on the internet. That gives step by step guide for installing

1. Spark
2. Anaconda python package that contains jupyter notebook
3. Setting up environment variables.

Some good tutorials are:

<https://medium.com/@naomi.fridman/install-pyspark-to-run-on-jupyter-notebook-on-windows-4ec2009de21f>

<https://acadgild.com/blog/spark-integration-with-jupyter-notebook-in-10-minutes>

**Running PySpark:**

After installing spark, Python and setting up environment variables you can run **Pyspark shell** by typing PySpark in cmd (command prompt) or Anaconda cmd or through jupyter notebook.

Run and follow the following steps in jupyter notebook

**Step 1:** Find spark in jupyter notebook and Import Spark libraries.

|  |
| --- |
| import findspark  findspark.init()  import pyspark  from pyspark.sql import \*  from pyspark import SparkContext, SparkConf |

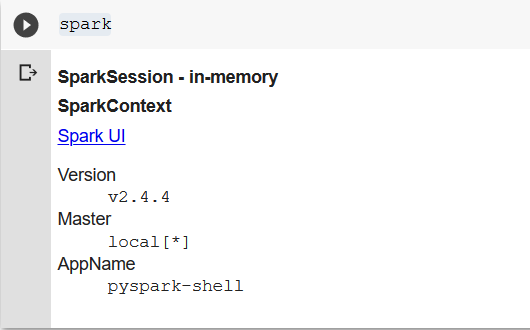
**Step 4: Create Spark session and Context.**

|  |
| --- |
| # create the session  conf = SparkConf().setMaster("local").setAppName("My app")  # create the context  sc = pyspark.SparkContext(conf=conf)  spark = SparkSession.builder.getOrCreate() |

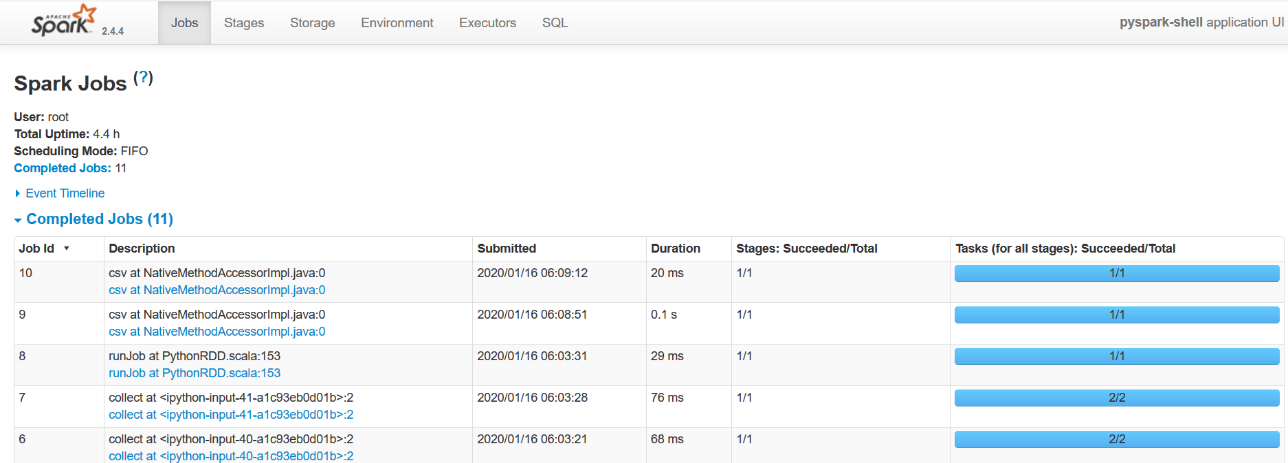
**Step 5: Run PySpark and view Spark Web interface.**

|  |
| --- |
| spark |

After executing this you can see the details. If you click on Spark UI it will take you to Spark web interface. In the Spark UI, you can monitor the progress of your job and debug the performance bottlenecks (if your Colab is running with a **local runtime**).



Click on **Spark UI** to open *(in case you are using a proxy then first switch it off)*



**Now, you can upload data to the notebook and start writing your PySpark code.**

|  |
| --- |
| nums = sc.parallelize([1,2,3,4])  nums.collect() |

|  |
| --- |
| inputfile = sc.textFile("F:\input.txt")  inputfile.count()  inputfile.first()  infile = inputfile.filter(lambda x: "and" in x)  infile.count()  output1 = inputfile.map(lambda line: line.split(" ")).filter(lambda line: len(line)>5)  output1.count()  output1.collect() |