HOW TO USE JUPYTER NOTEBOOKS WITH APACHE SPARK

In this article, we explain how to set up PySpark for your Jupyter notebook. This setup lets you write Python code to work with Spark in **Jupyter**.

Many programmers use Jupyter, formerly called iPython, to write Python code, because it's so easy to use and it allows graphics. Unlike Zeppelin notebooks, you need to do some initial configuration to use Apache Spark with Jupyter.

An important note on Apache clusters. You cannot use Jupyter with an Apache cluster because PySpark doesn't work with clusters. Luckily, you don't need that when working with Jupyter because it runs your jobs on whatever Spark instance you indicate. But Jupyter cannot run jobs across the cluster—it won't run the code in distributed mode. This is only an issue in very large data sets, in which case you'd use **submit-spark** to run your code on the cluster.

Now, let's get starting setting up PySpark for your Jupyter notebook.

Setting PySpark and Jupyter environment variables

First, all these environment variables. These set **PySpark** so that it will use that content and then pass it to the Jupyter browser.

Below, I use an IP address that's routable on an internal network, so that I can read my Jupyter notebook from the public internet. I put –no-browser so that it won't open a browser on my local device. If you only want to run this on your laptop, you can use the loopback address.

```
export SPARK_HOME='/usr/share/spark/spark-3.0.0-preview-bin-hadoop2.7'
```

```
export PATH=$PATH:$SPARK_HOME/bin

export PYSPARK_DRIVER_PYTHON="jupyter"

export PYSPARK_DRIVER_PYTHON_OPTS="notebook --no-browser --port=8889"

export SPARK_LOCAL_IP="172.31.46.15"
```

Now run **PySpark**. You will get a screen like this, below. Paste the pink URL into your browser.

(In the example, parisx is the internal address. So, I would replace it with the internet one, such as *mydomain.com:8889/?token=6cfc363cf7dab1f2e1f2c73b37113ef496155595b29baac5*)

```
 Serving notebooks from local directory: /home/ubuntu
 The Jupyter Notebook is running at:
 http://parisx:8889/?token=6cfc363cf7dab1f2e1f2c73b37113ef496155595b29baac5
 Use Control-C to stop this server and shut down all kernels (twice to skip
confirmation).

http://parisx:8889/?token=6cfc363cf7dab1f2e1f2c73b37113ef496155595b29baac5
    To access the notebook, open this file in a browser:
        file:///run/user/1000/jupyter/nbserver-30498-open.html
    Or copy and paste one of these URLs:
http://parisx:8889/?token=6cfc363cf7dab1f2e1f2c73b37113ef496155595b29baac5
 404 GET /api/kernels/a769e52d-eaf2-49f7-
b79b-4fe588a7bdd0/channels?session_id=fbab46a7332344e48d3052f36f6e589f
(71.12.95.23): Kernel does not exist: a769e52d-eaf2-49f7-b79b-4fe588a7bdd0
 404 GET /api/kernels/a769e52d-eaf2-49f7-
b79b-4fe588a7bdd0/channels?session_id=fbab46a7332344e48d3052f36f6e589f
(71.12.95.23) 30.85ms referer=None
 Replacing stale connection: a769e52d-eaf2-49f7-
b79b-4fe588a7bdd0:fbab46a7332344e48d3052f36f6e589f
```

If you want the notebook to keep running when you disconnect, use **nohup pyspark&** to run it as a background job. Then cat the file **nohup.out** to see the token number to use.

The code, explained

Below is sample code to prove that it works. Unlike the PySpark shell, when you use Jupyter you have to get the **SparkContext** and **SQLContext**, as shown below. You do not need to create the SQLContext; that is already done by PySpark.

```
sc = pyspark.SparkContext.getOrCreate(conf=conf)
sqlcontext = SQLContext(sc)
```

```
from pyspark.sql.types import StructType, StructField, FloatType, BooleanType
from pyspark.sql.types import DoubleType, IntegerType, StringType
import pyspark
```

```
from pyspark import SQLContext

conf = pyspark.SparkConf()

sc = pyspark.SparkContext.getOrCreate(conf=conf)
sqlcontext = SQLContext(sc)

schema = StructType()

data = ()
df=sqlcontext.createDataFrame(data,schema=schema)
```

Lastly, display the data.

```
df.show()

+-----+------------+
|sales|sales person|
+-----+------------+
|   10|      Walker|
|   20|     Stepher|
+-----+------------+
```