PySpark Windows Setup

The setup process is very simple, no need explicitly setup of Scala and Hadoop. Just follow these steps: -

1 Install Java:

Java is used by many other software. So, it is quite possible that a required version (in our case version 7 or later) is already available on your computer. To check if Java is available and find its version, open a Command Prompt and type the following command.

java -version

2 Install Anaconda (for python)

To check if Python is available, open a Anaconda Command Prompt and type the following command.

python --version

3 Create a virtual environment (optional)

conda create -n vpyspark python==3.6.8 **(creating new environment)**

conda activate vpyspark **(activating created environment)**

pip install jupyter notebook **(to run in notebook)**

4 Install Pyspark

pip install pyspark **(It will directly download the latest stable release of pyspark)**

pip install findspark

5 Path setup

JAVA_HOME = C:\Program Files\Java\jdk1.8.0_181 (in your case jdk may be different)

path = C:\ProgramData\Anaconda3\envs\vpyspark\Lib\site-packages\pyspark

path = C:\Program Files\Java\jre1.8.0_181\bin (in your case jre may be different)

PYSPARK_DRIVER_PYTHON = C:\ProgramData\Anaconda3\envs\vpyspark\Scripts\jupyter.exe

PYSPARK_DRIVER_PYTHON_OPTS = notebook

PYSPARK_PYTHON = C:\ProgramData\Anaconda3\envs\vpyspark\python.exe

SPARK_HOME = C:\ProgramData\Anaconda3\envs\vpyspark\Lib\site-packages\pyspark

6 Setup verification

spark-submit –version

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.5
      /_/

Using Scala version 2.11.12, Java HotSpot(TM) 64-Bit Server VM, 1.8.0_181
Branch HEAD
Compiled by user centos on 2020-02-02T19:38:06Z
Revision cee4ecbb16917fa85f02c635925e2687400aa56b
Url https://gitbox.apache.org/repos/asf/spark.git
Type --help for more information.
```

7 Open Notebook

Setup the path in Anaconda command prompt to work space and check environment as well.

Pyspark (it will directly open jupyter notebook)

Code	Notebook outcome

# Importing pyspark 	
import pyspark
# start the SparkContext 	
import findspark
findspark.init()
from pyspark import SparkContext
sc = SparkContext(master="local[2]")
# sc = SparkContext("local", "Simple\ App") 	
print(sc)
	`import pyspark`
	`# start the SparkContext`
	`import findspark`
	`findspark.init()`
	
	`from pyspark import SparkContext`
	`sc = SparkContext(master="local[2]")`
	`# sc = SparkContext("local", "Simple\ App")`
	`print(sc)`
	
	`<SparkContext master=local[2] appName=PySparkShell>`
	
	`sc.stop()`

Note:- sc can be allocated only ones in notebook terminal, so in order to allocate again sc.stop() need to be execute.

Github Link:- https://github.com/ravichaurasia