PySpark to run in Jupyter Notebook on Windows

1. Install Java 8

Before you can start with spark and hadoop, you need to make sure you have java 8 installed, or to install it.

Check if JAVA is installed

Open cmd (windows command prompt) , or anaconda prompt, from start menu and run:

```
java -version
```

You Should get something like:

```
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) Client VM (build 25.144-b01, mixed mode, sharing)
```

Check the setup for environment variables: JAVA_HOME and PATH, as described below.

Install JAVA 8

Download JAVA from Oracle website:

Run the executable, and JAVA by default will be installed in: C:\Program Files\Java\jdk1.8.0_201

Add the following **environment variable**:
JAVA_HOME = C:\Program Files\Java\jdk1.8.0_201

Add to PATH variable the following directory:
C:\Program Files\Java\jdk1.8.0_201\bin

2. Download and Install Spark

Go to Spark home page, and download the **.tgz** file from 2.3.2 version, according to time of writing, the pyspark in the latest version did not work correctly.

Extract the file to your chosen directory (7z can open tgz). In my case, it was C:\spark. There is another compressed directory in the tar, extract it (into here) as well.

Setup the environment variables

SPARK_HOME = C:\spark\spark-2.3.2-bin-hadoop2.7
HADOOP_HOME = C:\spark\spark-2.3.2-bin-hadoop2.7

Add the following path to PATH environment variable:
C:\spark\spark-2.3.2-bin-hadoop2.7\bin

3. Download and setup winutils.exe

- Goto https://github.com/steveloughran/winutils
- Choose your hadoop version, then go to bin
- Download the winutils.exe file.

Example: https://github.com/steveloughran/winutils/blob/master/hadoop-2.7.1/bin/winutils.exe

Save winutils.exe in to bin directory of your spark installation, **SPARK_HOME**bin directory.

In my case: C:\spark\spark-2.3.2-bin-hadoop2.7\bin.

4. Setup C:\tmp\hive directory

1. Create the folder **C:\tmp\hive**
2. Execute the following command in **cmd** started using the option **Run as administrator.**

   winutils.exe chmod -R 777 C:\tmp\hive
   winutils.exe ls -F C:\tmp\hive

   The output is something of the sort:
   drwxrwxrwx|1|LAPTOP-.....

5. Check PySpark installation

- In your anaconda prompt, or any python supporting cmd, type pyspark, to enter pyspark shell.
- To be prepared, best to check it in the python environment from which you run jupyter notebook. You supposed to see the following:

```
(py36) C:\Users\naomi>pyspark
Python 3.6.5 |Anaconda custom (64-bit)| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
2019-01-20 23:23:03 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-
java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0
      /_/

Using Python version 3.6.5 (default, Mar 29 2018 13:32:41)
SparkSession available as 'spark'.
>>>
```

Run the following commands at the python shell, the output should be [1,4,9,16].

```
>>> nums = sc.parallelize([1,2,3,4])
>>> nums.map(lambda x: x*x).collect()
```

To exit pyspark shell, type Ctrl-z and enter. Or the python command exit()

6. PySpark with Jupyter notebook

Install conda findspark, to access spark instance from jupyter notebook. Check current installation in Anaconda cloud.

In time of writing:

```
conda install -c conda-forge findspark
```

Open your python jupyter notebook, and write inside:

```
import findspark
findspark.init()
findspark.find()

import pyspark
findspark.find()
```

Last line will output SPARK_HOME path.

It's just for test, you can delete it.

```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSessionconf =
pyspark.SparkConf().setAppName('appName').setMaster('local')
sc = pyspark.SparkContext(conf=conf)
spark = SparkSession(sc)
```

Run the same test example as in pyspark shell:

```
nums = sc.parallelize([1,2,3,4])
nums.map(lambda x: x*x).collect()
```

In the end, stop the session

```
sc.stop()
```

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