Install Apache Spark Stand-alone

When you need to scale up your machine learning abilities, you will need a distributed computation. PySpark interface to Spark is a good option. Here is a simple guide, on installation of Apache Spark with PySpark, alongside your anaconda, on your windows machine.

Install Java 8

Before you can start with spark, you need to make sure you have java 8 installed, or to install it. First of all, check if JAVA is installed: open cmd (windows command prompt) , or anaconda prompt, from start menu and run:

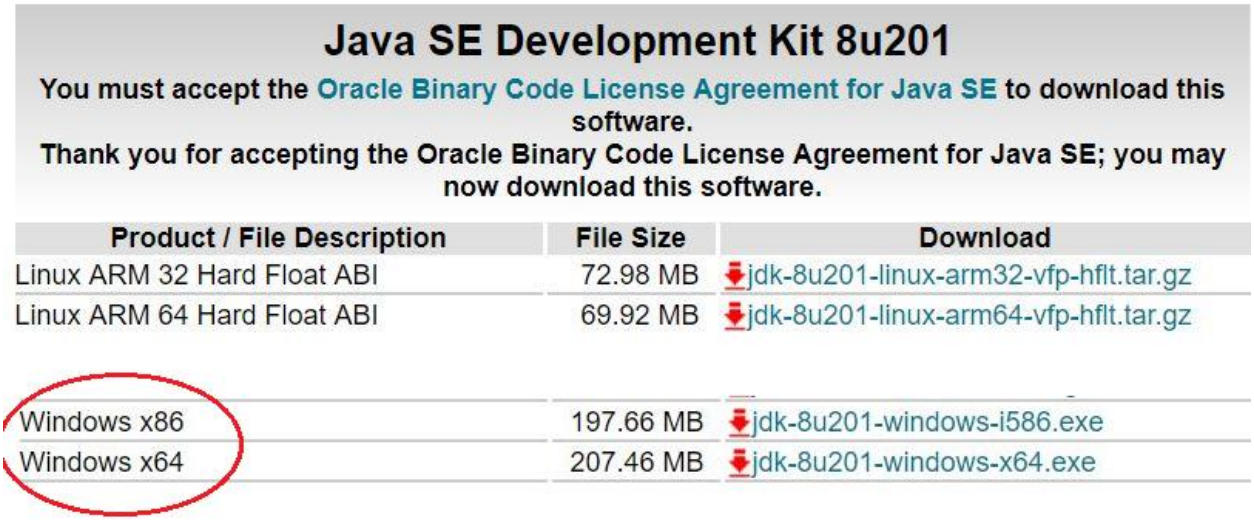
```
C:\>java -version
```

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)
```

If you don't have Java 8 installed, you need to install it.

Go to Java's official download website, accept Oracle license and download Java JDK 8, suitable to your system.



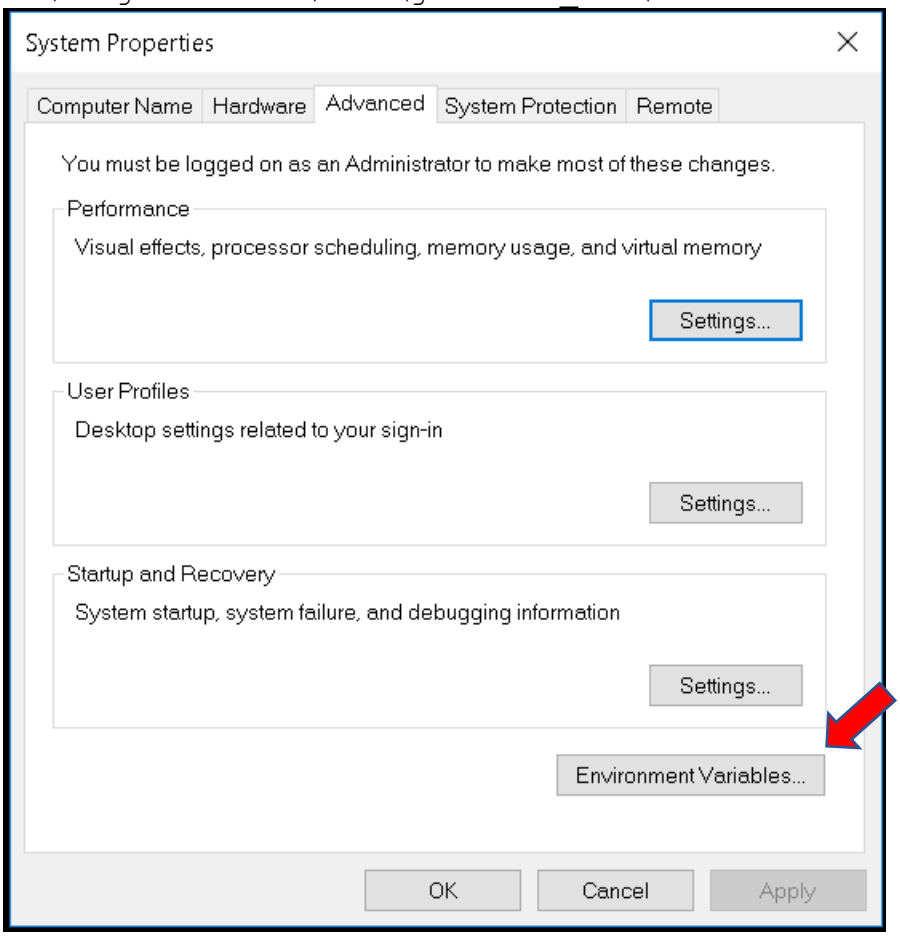
Java SE Development Kit 8u201

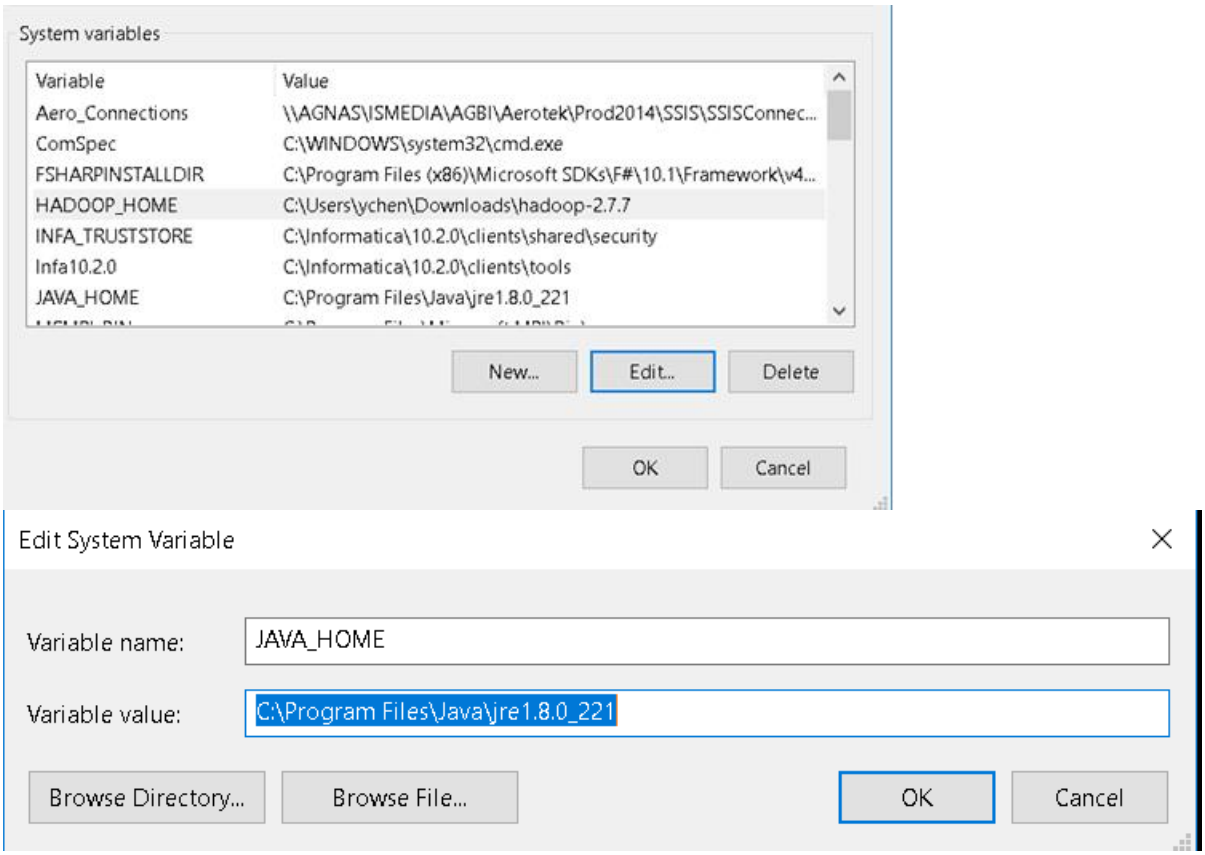
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.98 MB	jdk-8u201-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.92 MB	jdk-8u201-linux-arm64-vfp-hflt.tar.gz
Windows x86	197.66 MB	jdk-8u201-windows-i586.exe
Windows x64	207.46 MB	jdk-8u201-windows-x64.exe

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

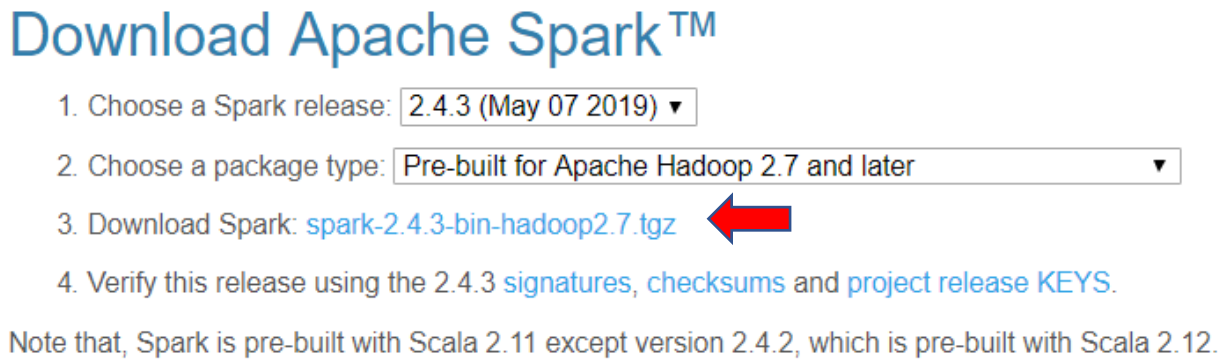
Add the following environment variable:
`JAVA_HOME = C:\Program Files\Java\jdk1.8.0_201`
Add to PATH variable the following directory:
`C:\Program Files\Java\jdk1.8.0_201\bin`





Download and Install Spark

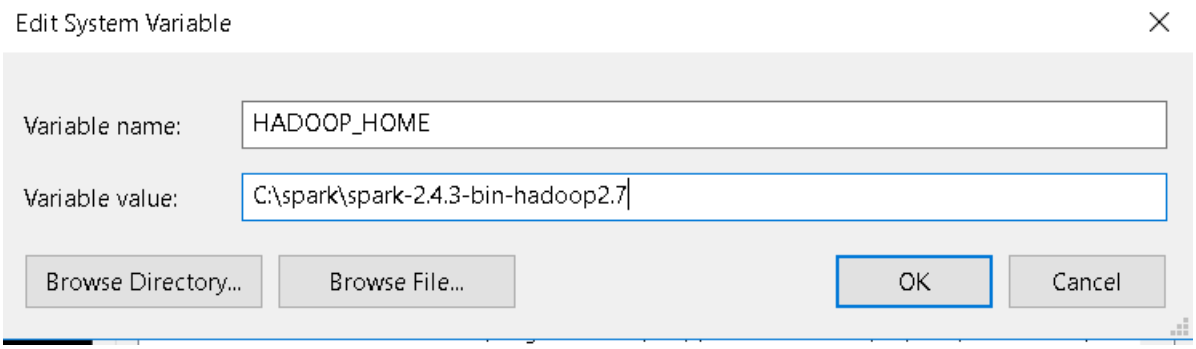
Go to [Spark home page](), and download the **.tgz** file from 2.4.3 version.



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

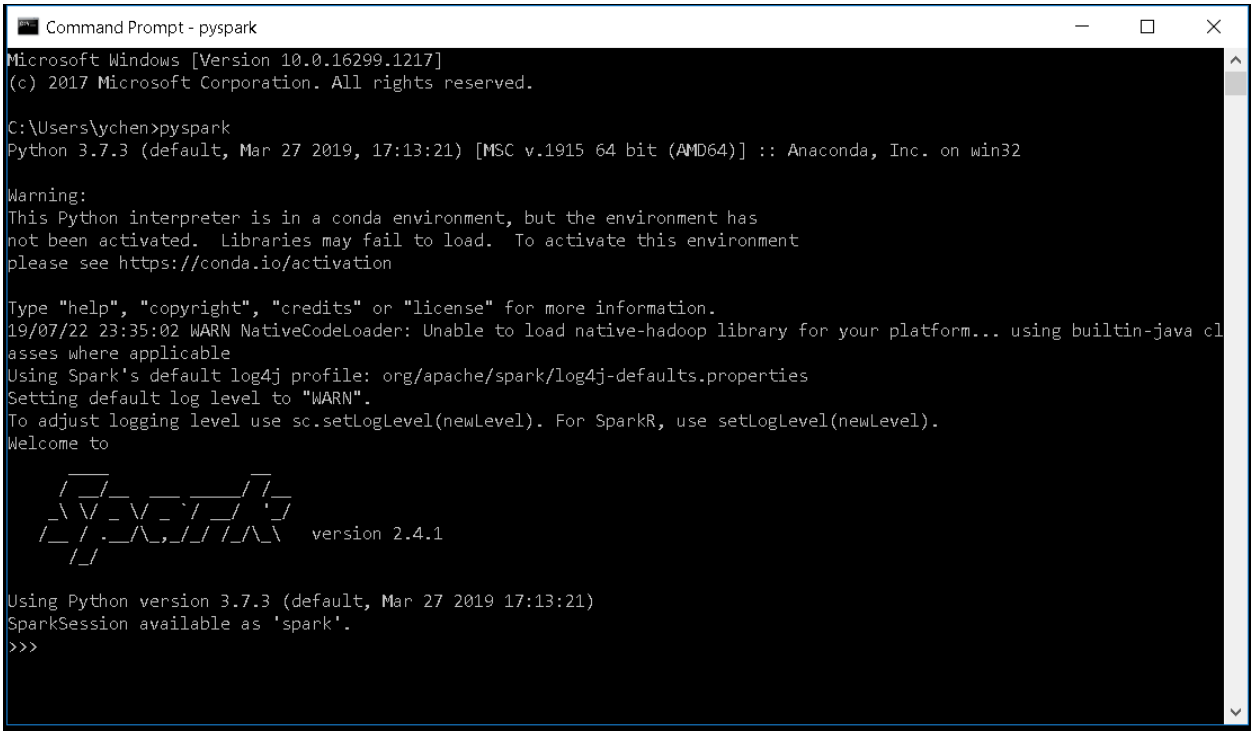
SPARK_HOME = C:\spark\spark-2.4.3-bin-hadoop2.7
HADOOP_HOME = C:\spark\spark-2.4.3-bin-hadoop2.7



Add the following path to PATH environment variable:
`C:\spark\spark-2.4.3-bin-hadoop2.7\bin`

Check PySpark Installation

In your anaconda prompt, or any python supporting cmd, type pyspark, to enter pyspark shell. You supposed to see the following:



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

Setup Hadoop with winutils.exe

In hadoop binaries repository of https://github.com/steveloughran/winutils, choose your hadoop version, then go to bin, and download the winutils.exefile. In my case: 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. Now the trick. It's not a must, things did not work well for me without it.

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

```
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-.....`

PySpark with Jupyter notebook

Install conda findspark to access spark instance from jupyter notebook. By writing:

```
C:\>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. The following lines will create a simple lamda function that can run parallelly on spark context.

```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSessionconf =
pyspark.SparkConf().setAppName('appName').setMaster('local')
sc = pyspark.SparkContext(conf=conf)
spark = SparkSession(sc)
nums = sc.parallelize([1,2,3,4])
nums.map(lambda x: x*x).collect()
sc.stop()
```

The following lines will test/show some important environment setup to use pyspark on jupyter notebook.

```
import os
print(os.environ['SPARK_HOME'])
print(os.environ['JAVA_HOME'])
print(os.environ['PATH'])
```