Data Science Toolkit on Windows 10

Installation Instruction

1. Install Anaconda

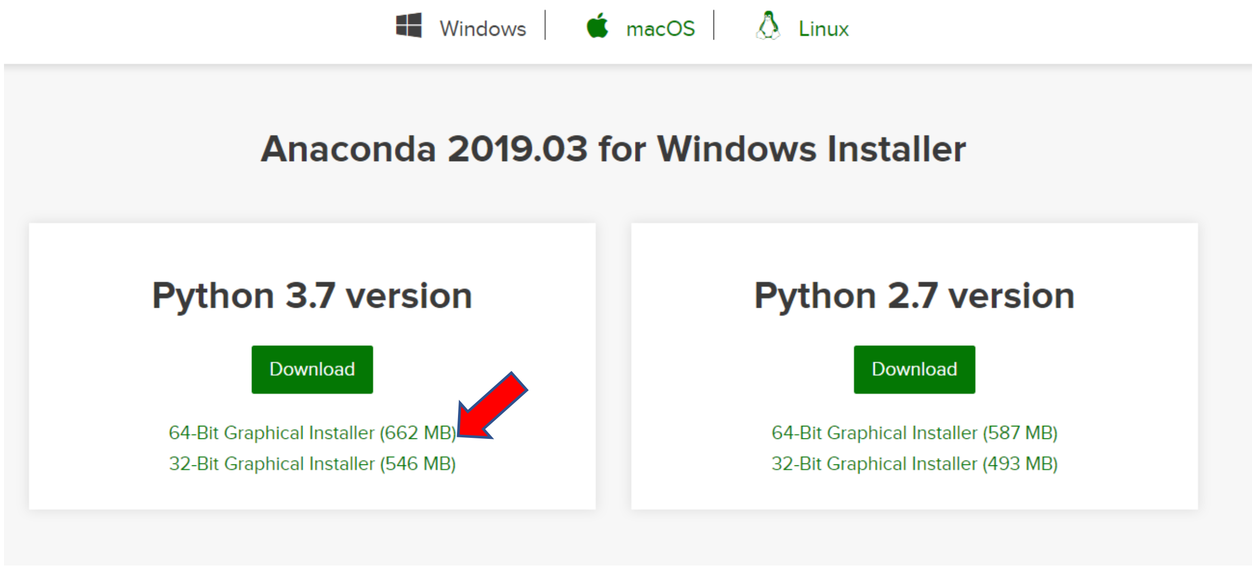
Designed for data science and machine learning workflows, Anaconda is an open-source package manager, environment manager, and distribution of the Python and R programming languages.

This section will guide you through installing Anaconda on a Windows 10 machine. Before installing software, you need to make sure you have administrator or other proper privileges, such as PowerBroker – Token Access Level 2.

Please Note: Uninstall Anaconda if you found your machine has an old version Anaconda. If you want to keep your existing Anaconda, you can ignore the first step of "Download and Install Anaconda" to avoid any conflictions.

Download and Install the Anaconda

From a web browser, go to the Anaconda Distribution page via the following link: https://www.anaconda.com/distribution/ to find the latest Windows version and download 64-bit Graphical Installer (assuming you have 64-bit computer).



Anaconda 2019.03 for Windows Installer

Python 3.7 version

64-Bit Graphical Installer (662 MB)
32-Bit Graphical Installer (546 MB)

Python 2.7 version

64-Bit Graphical Installer (587 MB)
32-Bit Graphical Installer (493 MB)

After downloading, locate "**Anaconda3-2019.03-Windows-x86_64.exe**" executable file and double-click to open it. And then Follow the instructions on the screen. If you are unsure about any setting, accept the defaults. You can change them later. ...

Test Installation

Use the `conda` command to test the installation and activation: For example,

```
C:\Users\ychen\anaconda3\bin>conda list
```

You'll receive output of all the packages you have available through the Anaconda installation.

Update Installation (Optional)

You can easily update Anaconda to the latest version.

```
C:\Users\ychen\anaconda3\bin>conda update --all --yes
```

Additional Installation of Python Libraries (Optional)

You can easily install some python popular data science libraries, such as tensorflow, keras, etc. by using conda install or pip install. For example, the following commands will install those useful libraries:

```
C:\Users\ychen\anaconda3\bin>pip install tensorflow pymc3 keras

C:\Users\ychen\anaconda3\bin>pip install fbprophet

C:\Users\ychen\anaconda3\bin>pip install clarify

C:\Users\ychen\anaconda3\bin>pip install pandas-profiling

C:\Users\ychen\anaconda3\bin>pip install spark-df-profiling

C:\Users\ychen\anaconda3\bin>pip install koalas

C:\Users\ychen\anaconda3\bin>pip install ipython-sql

C:\Users\ychen\anaconda3\bin>pip install jupyter_contrib_nbextensions

C:\Users\ychen\anaconda3\bin>jupyter contrib nbextension install --sys-
prefix

C:\Users\ychen\anaconda3\bin>pip install autopep8
```

You can use conda and pip to manage many python libraries/packages. The official documentation can be found here.

Install R Kernel and R Packages

To use R in an anaconda environment, all you need to do is to install the r-essentials bundle, which includes over 80 of the most popular scientific R packages.

```
C:\Users\ychen\anaconda3\bin>conda install r-essentials

C:\Users\ychen\anaconda3\bin>conda install r-irkernel
```

The R language packages are available to install with conda at http://repo.anaconda.com/pkgs/r/. You can install any of these R language packages into your current environment with the conda command `conda install -c r package-name`. For more information, you can check this link.

Install Scala Kernel for Jupyter Notebook

To use Scala in an anaconda environment, you can install Scala kernel for jupyter notebook as below:

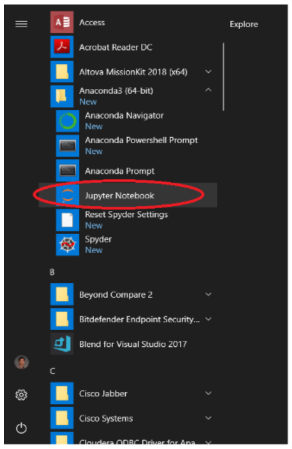
```
C:\Users\ychen\anaconda3\bin>conda install spylon-kernel
```

```
C:\Users\ychen\anaconda3\bin>python -m spylon_kernel install
```

There are a large number of kernels that will run within Jupyter Notebooks, as listed here.

Start Jupyter Notebook

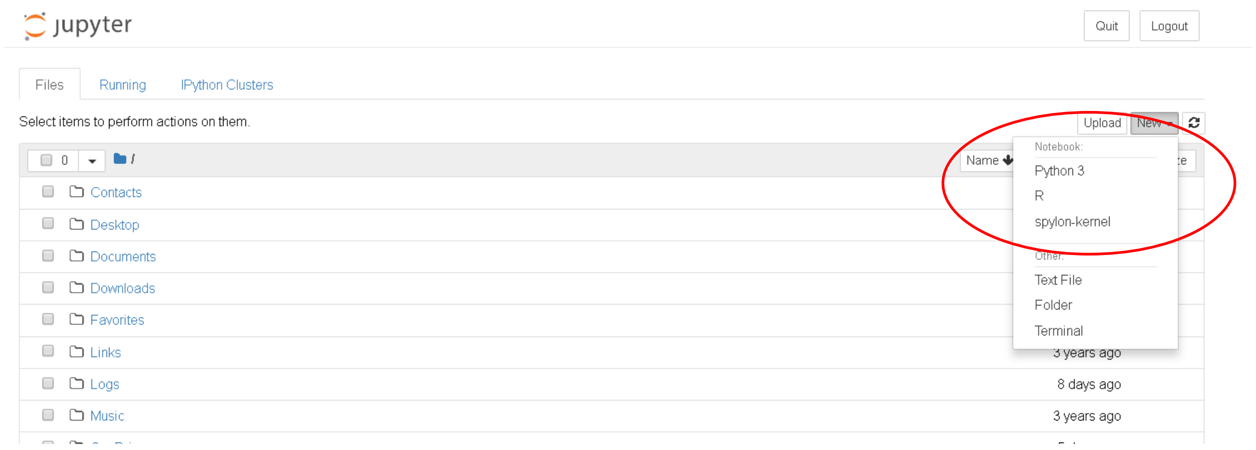
You can launch Jupyter Notebook from Start→Anaconda (64 bit)→Jupyter Notebook like following screen shot:



Or type the command from **cmd** windows prompt to start Jupyter Notebook:

```
C:\Users\ychen>jupyter notebook
```

If Jupyter notebook cannot be open in a web browser, you can copy the server url (e.g. http://localhost:8888/?token=0fa79120798f795452ebc143ce11d7f1f8ac73e5f860d551) and paste it on a open web browser, such as google chrome. After you can access the jupyter notebook web page, you should see the content like the following screen shots.



2.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:

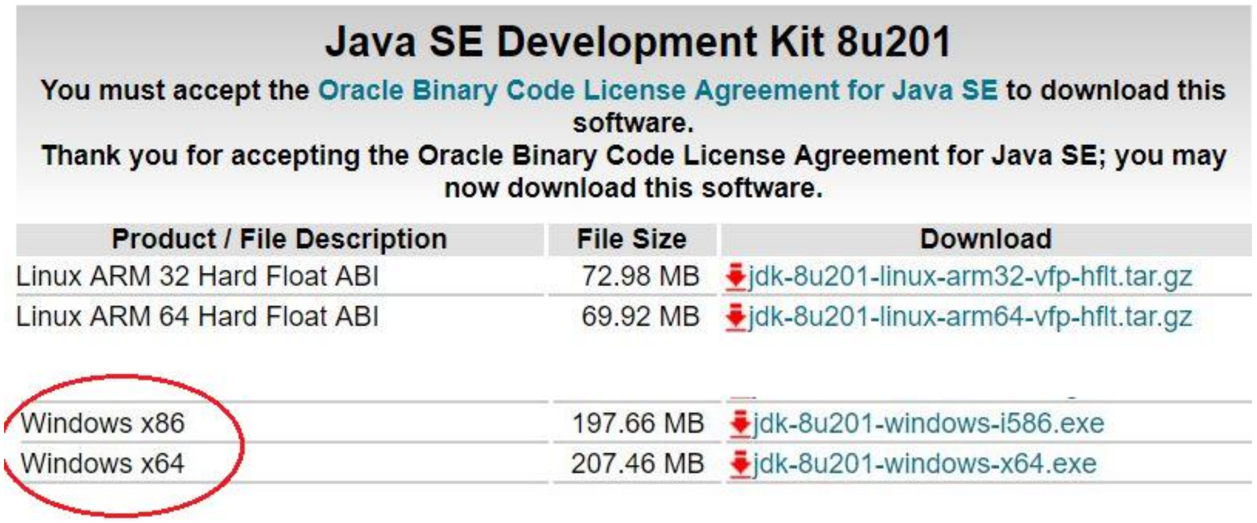
```
C:\Users\ychen>java -version
```

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

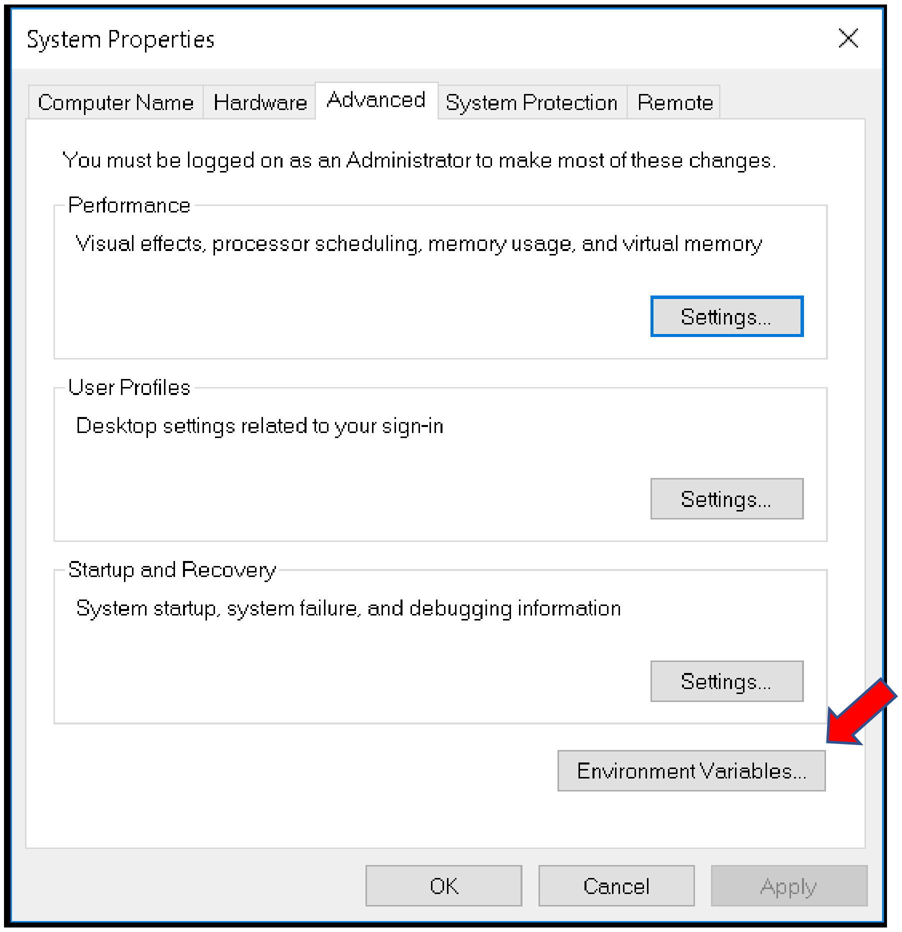
Run the executable, and JAVA by default will be installed in C:\Program Files\Java\jdk1.8.0_201

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
```


System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

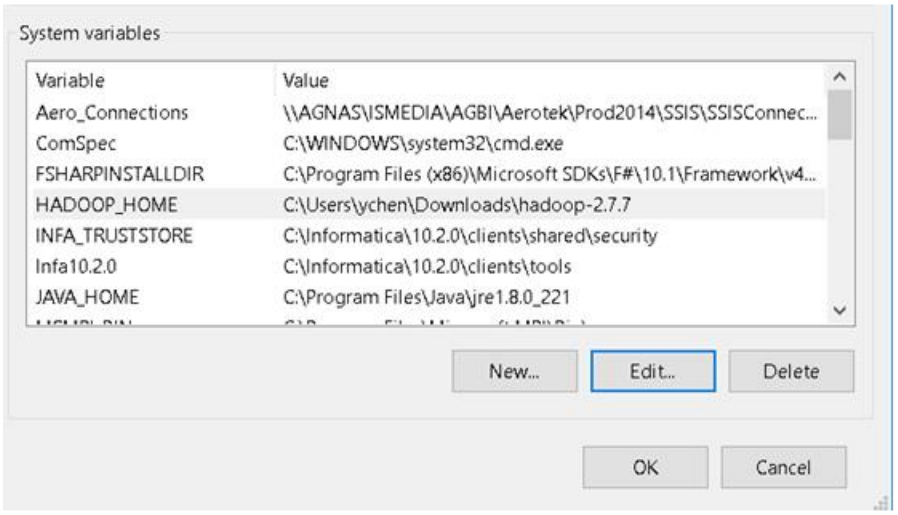
Startup and Recovery

System startup, system failure, and debugging information

Settings...

Environment Variables...

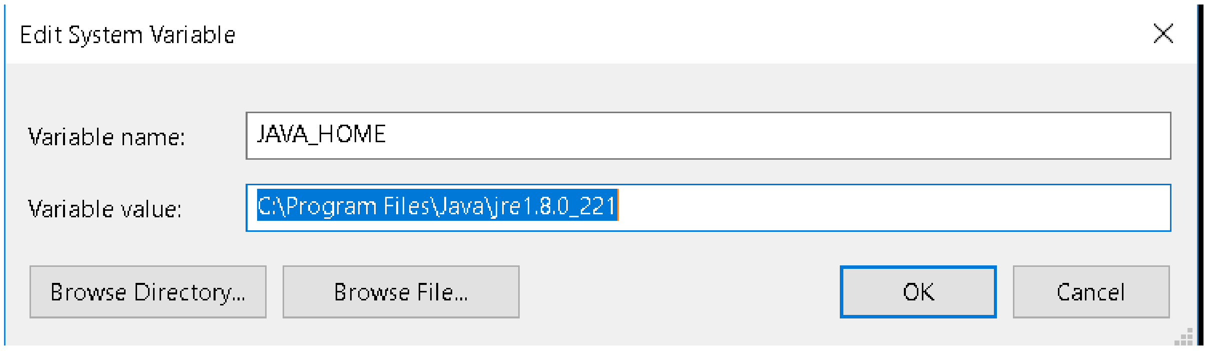
OK | Cancel | Apply


System variables

Variable	Value
Aero_Connections	\\AGNAS\ISMEDIA\AGBI\Aerotek\Prod2014\SSIS\SSISConnec...
ComSpec	C:\WINDOWS\system32\cmd.exe
FSHARPINSTALLDIR	C:\Program Files (x86)\Microsoft SDKs\F#\10.1\Framework\v4...
HADOOP_HOME	C:\Users\ychen\Downloads\hadoop-2.7.7
INFA_TRUSTSTORE	C:\Informatica\10.2.0\clients\shared\security
Infa10.2.0	C:\Informatica\10.2.0\clients\tools
JAVA_HOME	C:\Program Files\Java\jre1.8.0_221

New... | Edit... | Delete

OK | Cancel

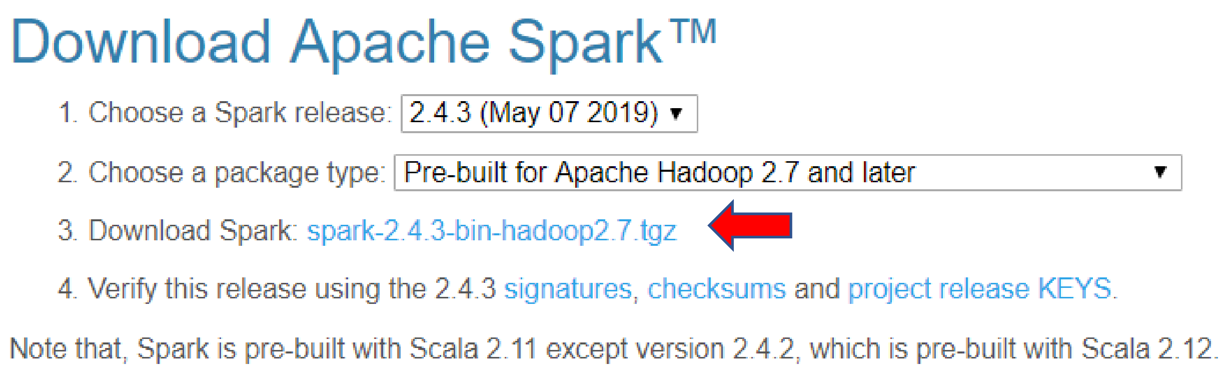

Edit System Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jre1.8.0_221

Download and Install Spark

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


Download Apache Spark™

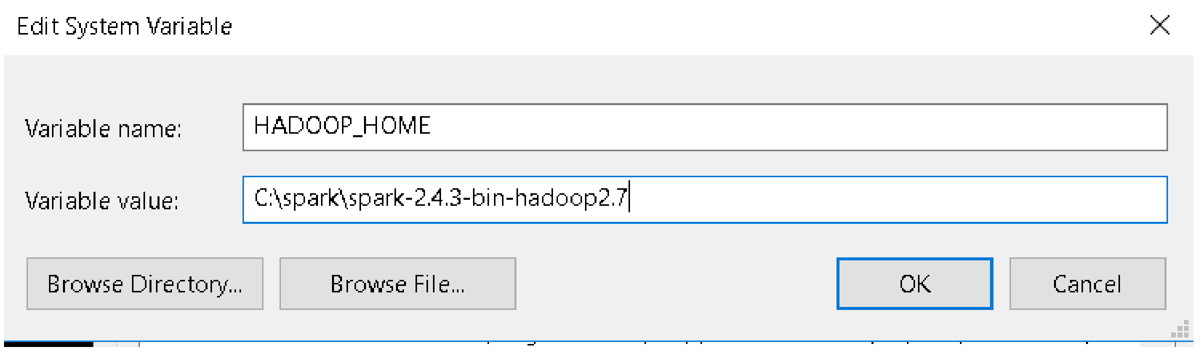
1. Choose a Spark release: 2.4.3 (May 07 2019) ▾
2. Choose a package type: Pre-built for Apache Hadoop 2.7 and later ▾
3. Download Spark: spark-2.4.3-bin-hadoop2.7.tgz
4. Verify this release using the 2.4.3 signatures, checksums and project release KEYS.

Note that, Spark is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12.

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
```


Edit System Variable

Variable name: HADOOP_HOME

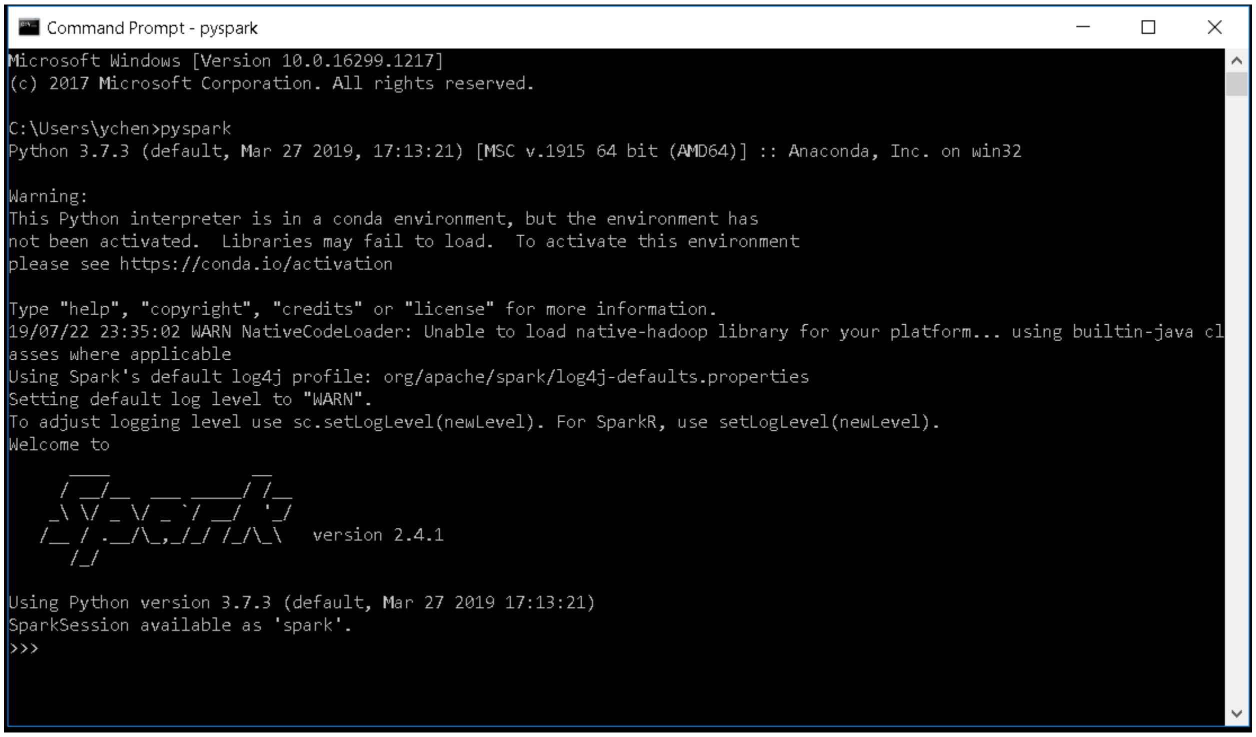
Variable value: C:\spark\spark-2.4.3-bin-hadoop2.7

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:



```
Microsoft Windows [Version 10.0.16299.1217]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ychen>pyspark
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
19/07/22 23:35:02 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.1
      /_/

Using Python version 3.7.3 (default, Mar 27 2019 17:13:21)
SparkSession available as 'spark'.
>>>
```

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

3.Setup Hadoop with winutils.exe

In hadoop binaries repository, 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-.....
```

4.PySpark with Jupyter notebook

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

```
C:\Users\ychen>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'])
```