Apache Spark Deployment

Trainer: Udayakumar M

Data Engineering

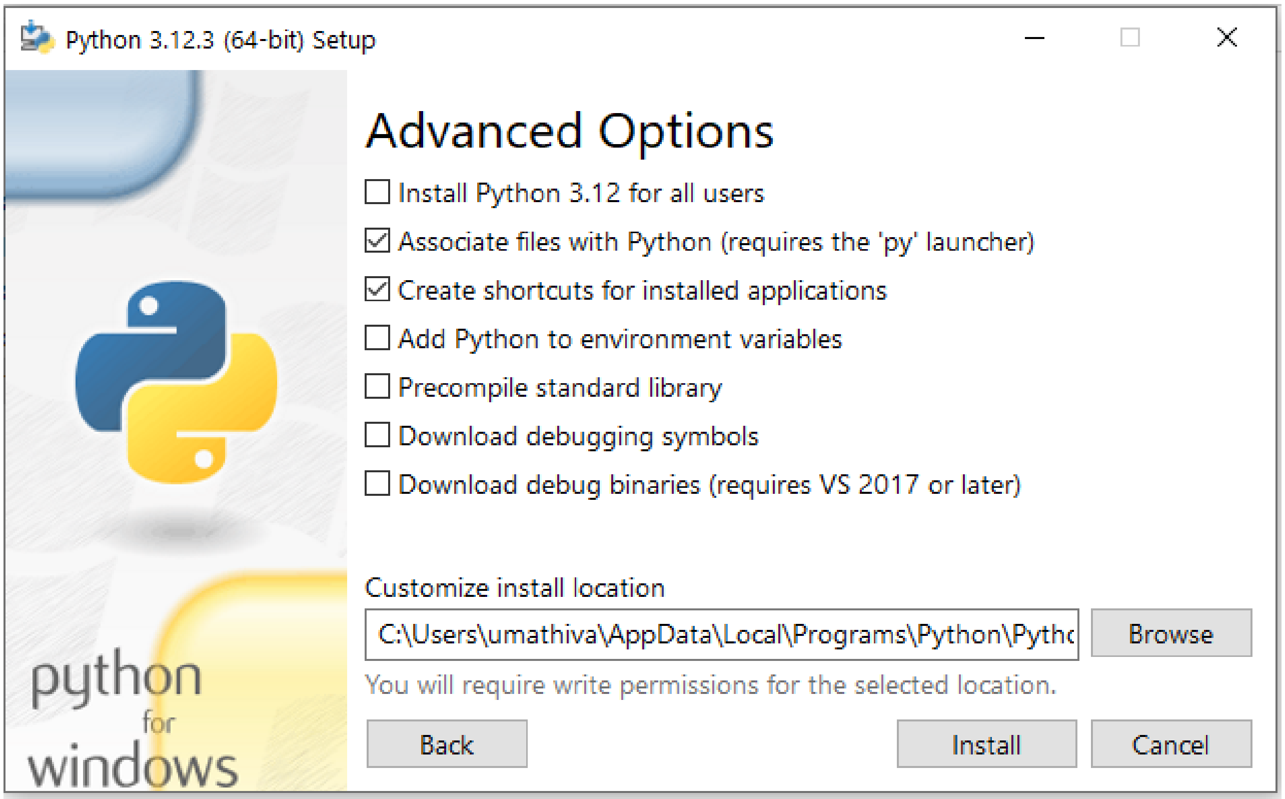
Introduction

The spark applications can be developed and executed on a spark cluster, to run/execute the spark applications first you must download, extract and copy the Apache Spark, and Hadoop to your local development system. Please follow the below instructions to run Spark applications on your local machine.

Apache Spark Installation and Configuration

Pre-requisite: Ensure to download and install the latest version of python on your local machine (make sure to choose/add Python related variables)

https://www.python.org/ftp/python/3.12.3/python-3.12.3-amd64.exe (Make sure to choose custom installation to get this page where you can choose the '**Add Python to environment variables**)



Python 3.12.3 (64-bit) Setup

Advanced Options

☐ Install Python 3.12 for all users

☑ Associate files with Python (requires the 'py' launcher)

☑ Create shortcuts for installed applications

☐ Add Python to environment variables

☐ Precompile standard library

☐ Download debugging symbols

☐ Download debug binaries (requires VS 2017 or later)

Customize install location

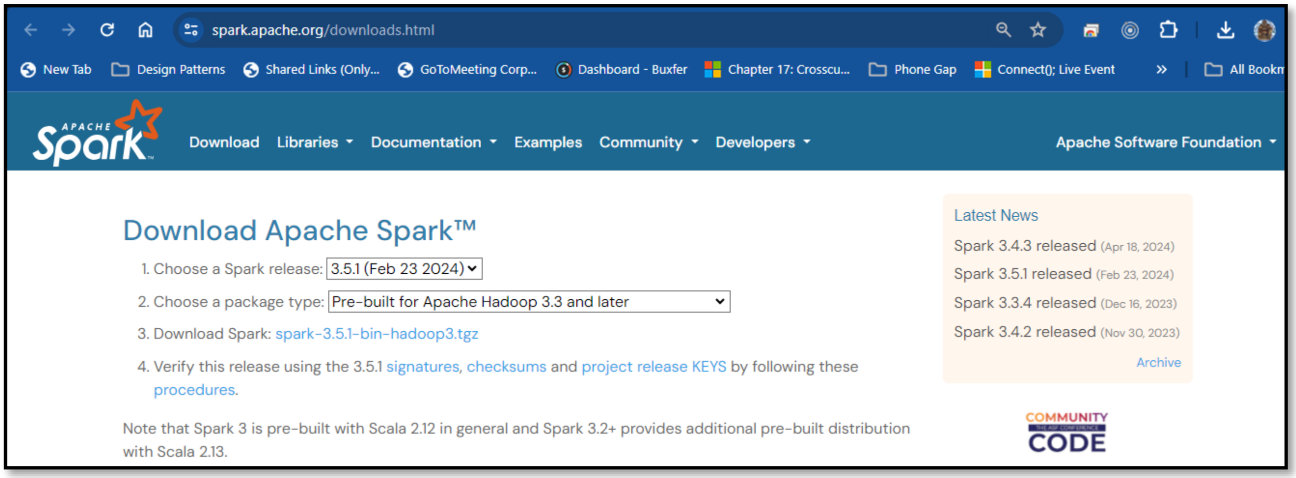
C:\Users\umathiva\AppData\Local\Programs\Python\Pytho

You will require write permissions for the selected location.

Step 1:

Download the Apache spark (version 3.5.1) from the below location.

https://spark.apache.org/downloads.html



Download Apache Spark™

1. Choose a Spark release: 3.5.1 (Feb 23 2024)
2. Choose a package type: Pre-built for Apache Hadoop 3.3 and later
3. Download Spark: spark-3.5.1-bin-hadoop3.tgz
4. Verify this release using the 3.5.1 signatures, checksums and project release KEYS by following these procedures.

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Latest News

Spark 3.4.3 released (Apr 18, 2024)

Spark 3.5.1 released (Feb 23, 2024)

Spark 3.3.4 released (Dec 16, 2023)

Spark 3.4.2 released (Nov 30, 2023)

Archive

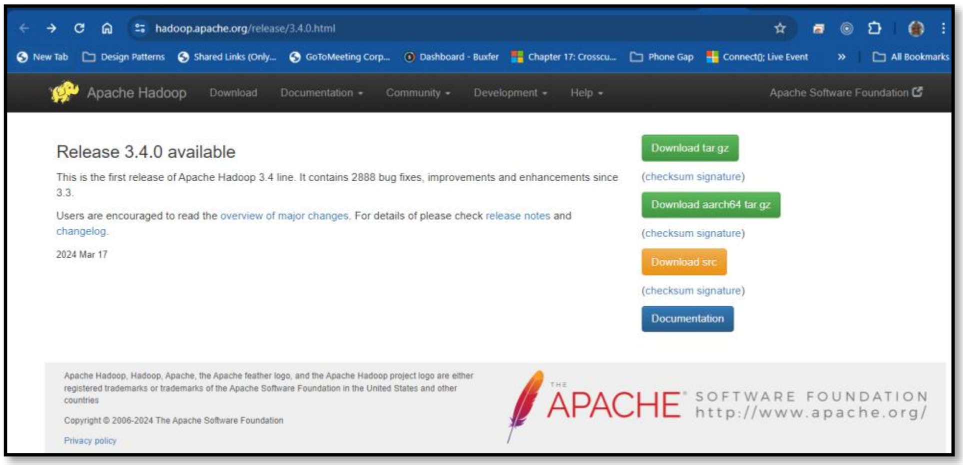
https://www.apache.org/dyn/closer.lua/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz

Extract the **.tgz** file using (winrar utility) into a folder something like **C:\Spark**

Step 2:

Download Hadoop (version 3.4.0) from the below location.

https://hadoop.apache.org/release/3.4.0.html



Release 3.4.0 available

This is the first release of Apache Hadoop 3.4 line. It contains 2888 bug fixes, improvements and enhancements since 3.3.

Users are encouraged to read the overview of major changes. For details of please check release notes and changelog.

2024 Mar 17

(checksum signature)

(checksum signature)

(checksum signature)

Apache Hadoop, Hadoop, Apache, the Apache feather logo, and the Apache Hadoop project logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and other countries

Copyright © 2006-2024 The Apache Software Foundation

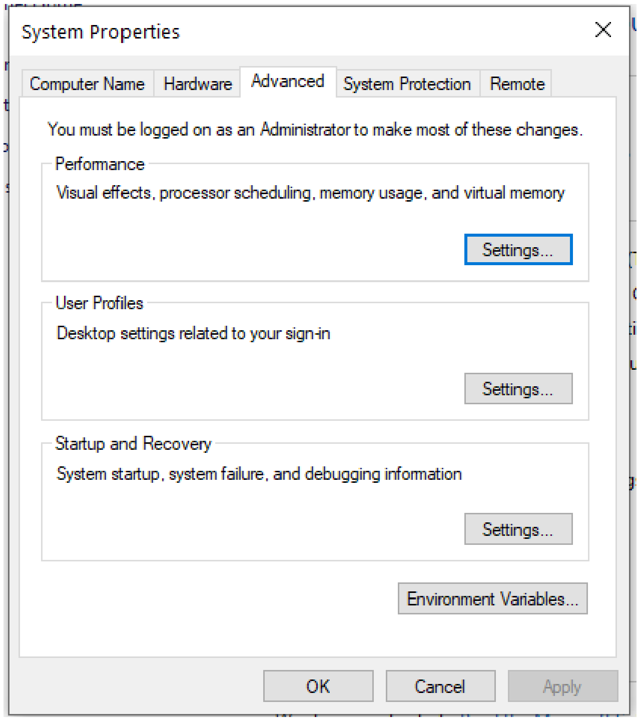
Privacy policy

https://archive.apache.org/dist/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

Extract the **.tgz** file using (winrar utility) into a folder like **C:\Hadoop**

Step 3:

Now add the below environment variables (**user variables**) to your system



System Properties

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

Performance

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

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

HADOOP_HOME: Despite Spark can run without Hadoop, the version we downloaded is prebuilt for Hadoop 3.3 and looks in the code for it (**C:\Hadoop)**. To fix this inconvenience we need to set this variable to the folder containing the winutils.exe file.

HADOOP_HOME = C:\Spark\hadoop

JAVA_HOME: usually this variable should be already set when you install java, but it is better to verify that exists and is correct.

JAVA_HOME = C:\Program Files\Java\jdk-12.0.2

SPARK_HOME: This variable should contain the path of the bin folder of where you uncompressed Spark .tgz file which is **C:\Spark**

SPARK_HOME = C:\Spark\spark-3.4.3-bin-hadoop3

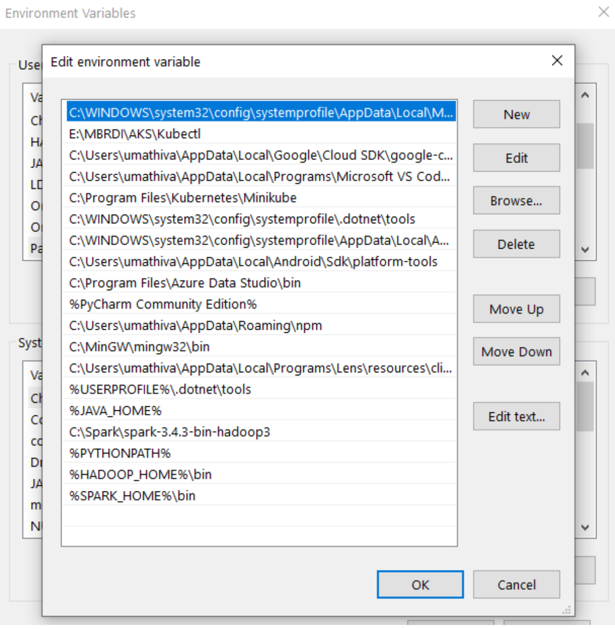
After the above variables are set, now add the below paths to PATH variable.

%JAVA_HOME%

%PYTHONPATH%

%HADOOP_HOME%\bin

%SPARK_HOME%\bin



```
C:\WINDOWS\system32\config\systemprofile\AppData\Local\M...
E:\MBRDI\AKS\Kubectl
C:\Users\umathiva\AppData\Local\Google\Cloud SDK\google-c...
C:\Users\umathiva\AppData\Local\Programs\Microsoft VS Cod...
C:\Program Files\Kubernetes\Minikube
C:\WINDOWS\system32\config\systemprofile\.dotnet\tools
C:\WINDOWS\system32\config\systemprofile\AppData\Local\A...
C:\Users\umathiva\AppData\Local\Android\Sdk\platform-tools
C:\Program Files\Azure Data Studio\bin
%PyCharm Community Edition%
C:\Users\umathiva\AppData\Roaming\npm
C:\MinGW\mingw32\bin
C:\Users\umathiva\AppData\Local\Programs\Lens\resources\cli...
%USERPROFILE%\.dotnet\tools
%JAVA_HOME%
C:\Spark\spark-3.4.3-bin-hadoop3
%PYTHONPATH%
%HADOOP_HOME%\bin
%SPARK_HOME%\bin
```

Now to develop and run your spark programs from VS Code Jupyter Notebook add the below environment variables

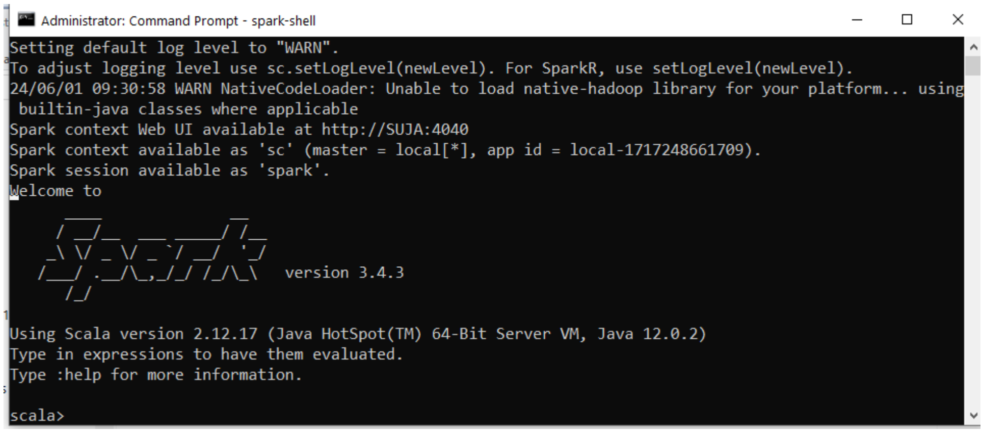
PYSPARK_PYTHON – Specify the path of the Python location where you have installed.

PYSPARK_DRIVER_PYTHON – Specify the value '**jupyter**' here.

PYSPARK_DRIVER_PYTHON_OPTS – Specify the value '**notebook**' here.

We are finally done and could start the spark-shell which is an interactive way to analyze data using Scala or Python. In this way we are going also to test our Spark installation.

From the command prompt go to **the Spark f**older and type the following command to run the Scala shell:

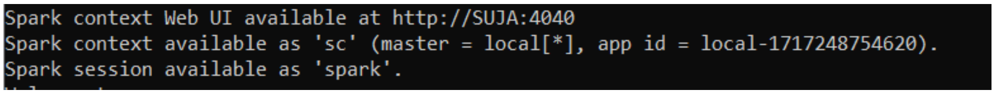


```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/06/01 09:30:58 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
 builtin-java classes where applicable
Spark context Web UI available at http://SUJA:4040
Spark context available as 'sc' (master = local[*], app id = local-1717248661709).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.4.3
      /_/

Using Scala version 2.12.17 (Java HotSpot(TM) 64-Bit Server VM, Java 12.0.2)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Notice the Spark Context and Spark Session got created successfully, optionally you shall navigate to http://127.0.0.1:4040 where Spark context web UI will be available.



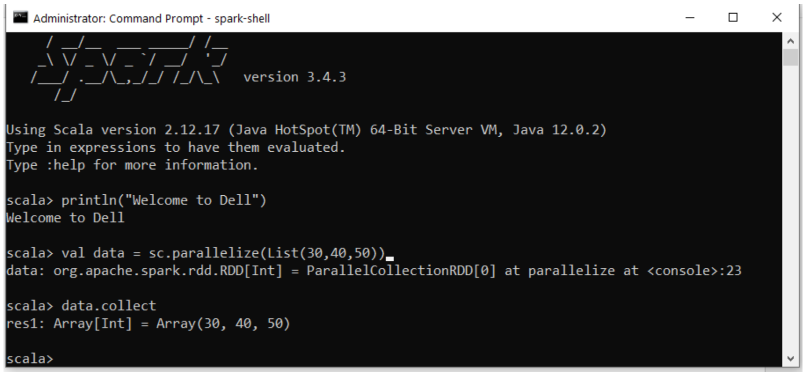
```
Spark context Web UI available at http://SUJA:4040
Spark context available as 'sc' (master = local[*], app id = local-1717248754620).
Spark session available as 'spark'.
```

Now run few scala commands to check the spark results

scala>println(“Welcome to Dell”)

scala>val data = sc.parallelize(List(30,40,50))

scala>data.collect



```
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.4.3
      /_/

Using Scala version 2.12.17 (Java HotSpot(TM) 64-Bit Server VM, Java 12.0.2)
Type in expressions to have them evaluated.
Type :help for more information.

scala> println("Welcome to Dell")
Welcome to Dell

scala> val data = sc.parallelize(List(30,40,50))
data: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:23

scala> data.collect
res1: Array[Int] = Array(30, 40, 50)

scala>
```

Hurray, you have successfully deployed Spark on your machine and ready to run spark applications in your local machine.