01 Installation - Handout -KirkYagami 👨‍💻 🦹‍♀️

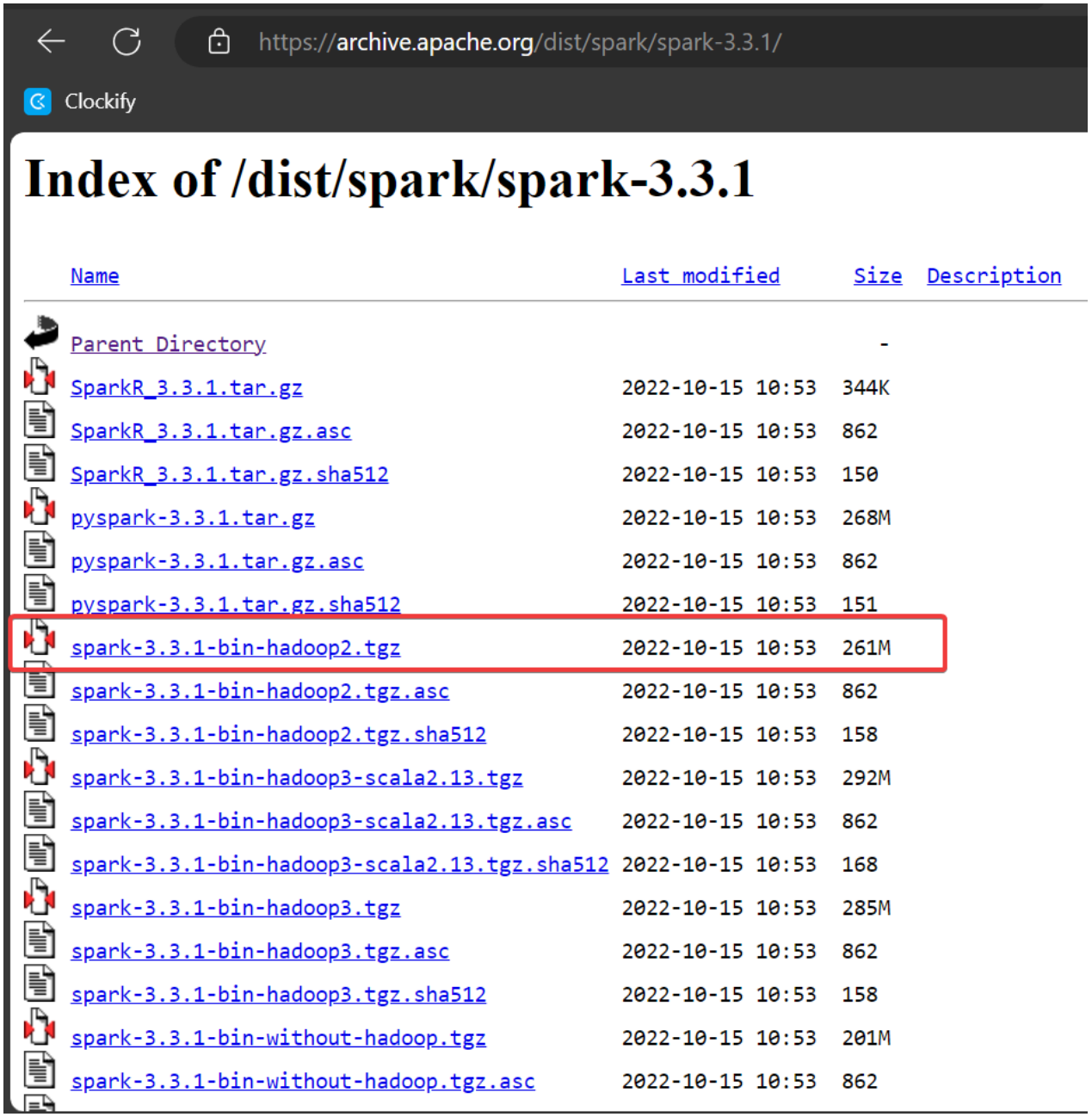
Instruction: Read this document at least two times and then assess yourself whether you can follow these installation steps.

Requirements

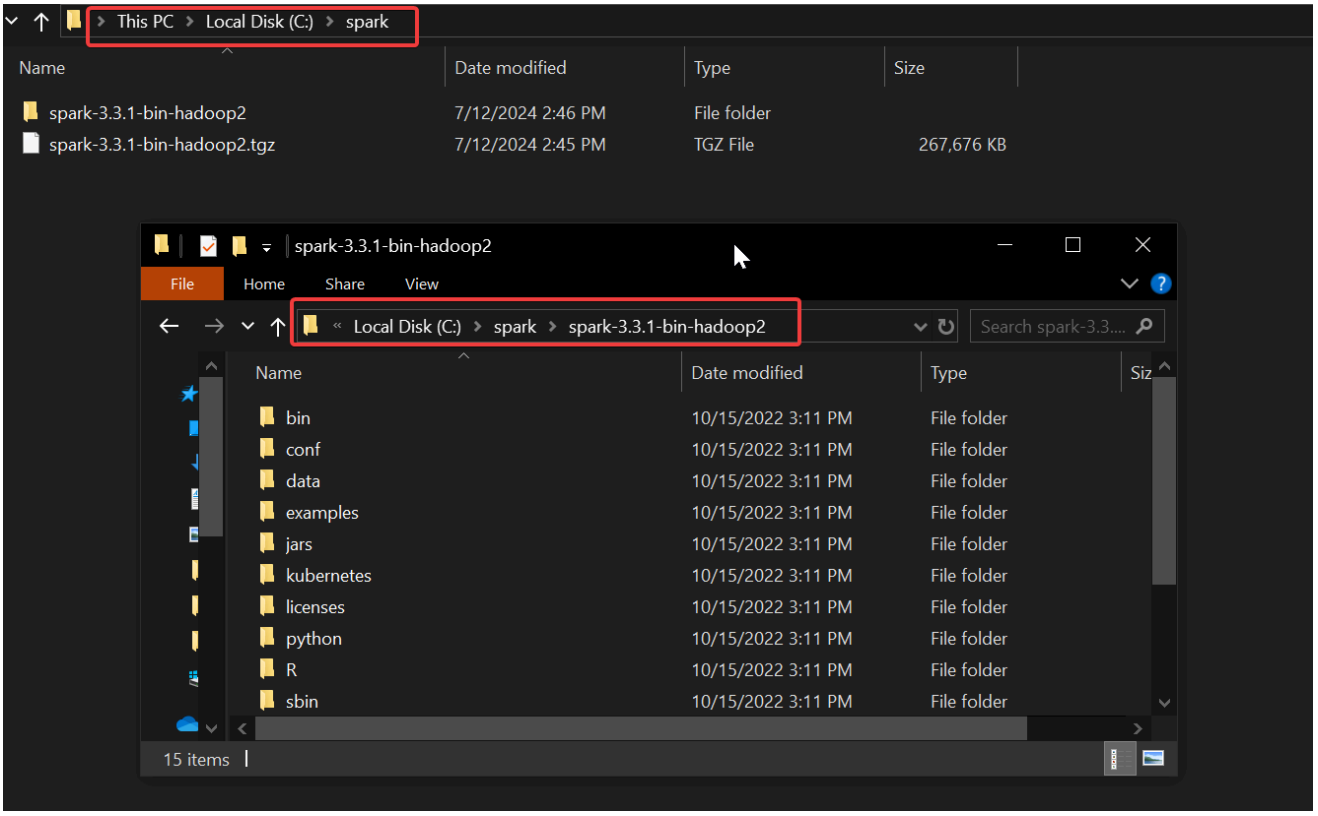
1. Java 19
2. Python ~~latest~~ 3.10.1 (Yes, it is little old but you will not miss anything. Don't Worry!)
3. spark 3.3.1 for hadoop 2.7

---

1. Java
   Java Archive Downloads - Java SE 19 (oracle.com)
   - Download the windows x64 installer file
   - install it when asked to choose path - choose `C:\Java\jdk` (Go to your C Drive create a folder called Java and inside it create another folder jdk)
   - Done!
2. Python
   https://www.python.org/downloads/
   - ~~Just install the latest python exe and enable path.~~
   - Install Python 3.10.1 because spark 3.3.1 is not compatible with later versions
   - https://www.python.org/downloads/release/python-3101/
3. spark
   Index of /dist/spark/spark-3.3.1 (apache.org)



4. Create a folder called `spark` in your C drive
5. Cut and paste the downloaded file in `C:\spark` and extract it there
6. Tip: If you move(cut and paste) the downloaded tgz file it will save you some time, and extract it inside the spark folder in C drive
7. If you are using 7-Zip to unzip the downloaded file, you will have to unzip it two times, and then move the content to just parent folder and delete the empty folder.

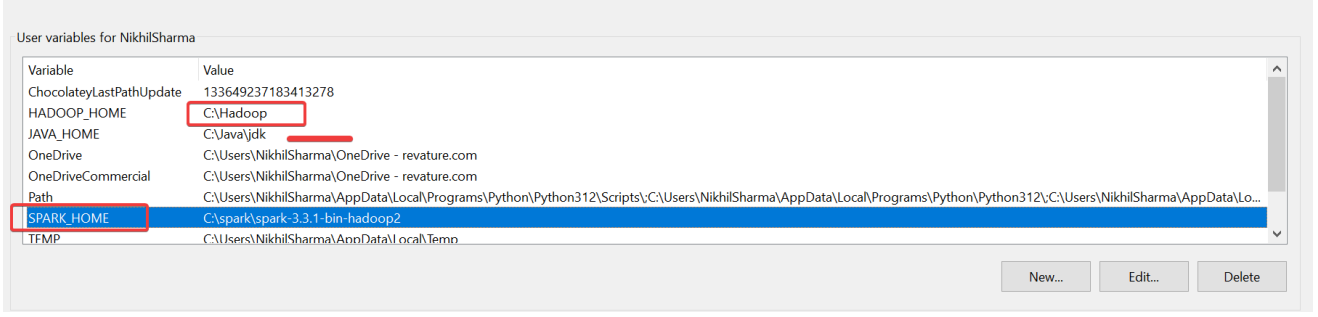


5. Hadoop Home
   https://github.com/steveloughran/winutils/blob/master/hadoop-3.0.0/bin/winutils.exe
   Download this file and put it in `C:\Hadoop\bin`

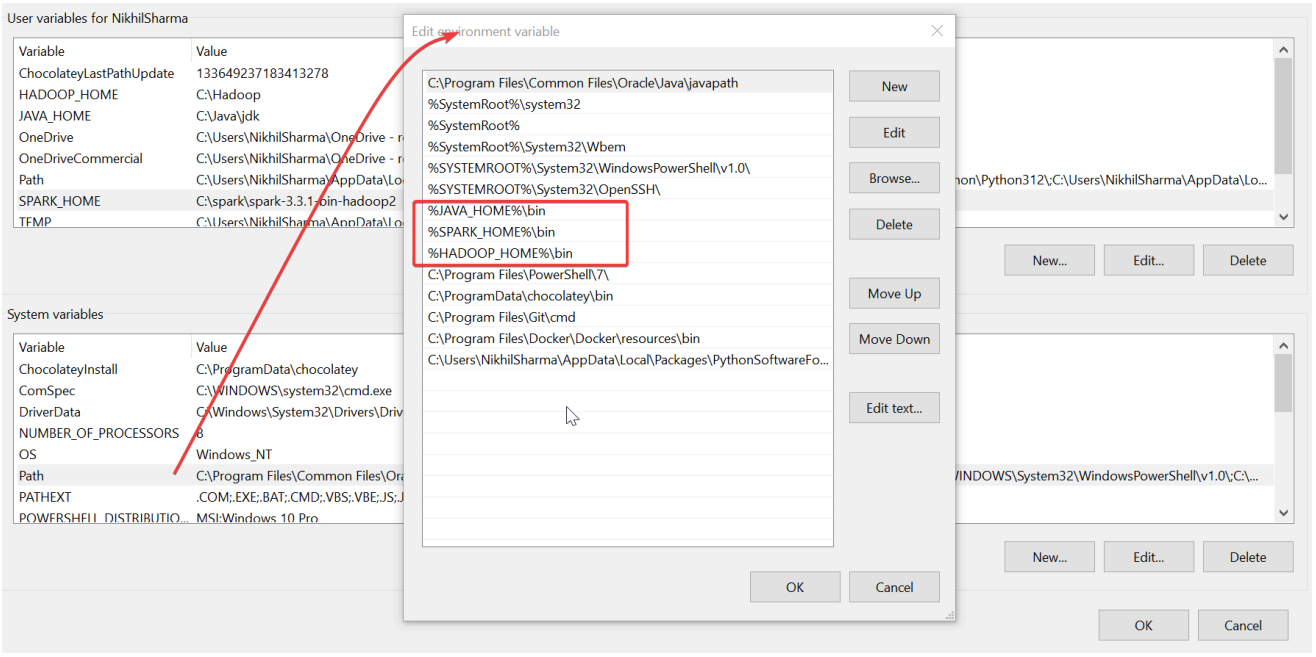
-- Final Step is creating these Environment variables

USER Variables

```
HADOOP_HOME- C:\hadoop
JAVA_HOME- C:\java\jdk
SPARK_HOME- C:\spark\spark-3.3.1-bin-hadoop2
```



System PATH



Execute below commands

```
pip install py4j
pip install pyspark==3.3.1
```

```
PS C:\Users\NikhilSharma> pyspark
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/07/12 15:47:59 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
s where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024 19:30:16)
Spark context Web UI available at http://host.docker.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1720779480395).
SparkSession available as 'spark'.
>>>
```

If it worked as expected you should be able to run PySpark in your machine!!

Now you ahead and install Jupyter Lab.

Run the below command in your terminal.

```
pip install jupyter notebook jupyterlab
```

`cd` to a folder of your choice and run `jupyter lab`

```
jupyter lab
```

This will open Jupyter lab in a browser tab, create a new `ipynb` file and run the below code.

Spark DataFrame

```
from pyspark.sql import SparkSession
```

```
# Create a SparkSession
spark = SparkSession.builder \
    .appName("DataFrameExample") \
    .getOrCreate()

# Create a DataFrame from a list of tuples
data = [("John", 25), ("Alice", 30), ("Bob", 35)]
df = spark.createDataFrame(data, ["Name", "Age"])

# Show the DataFrame
df.show()

# Filter the DataFrame
filtered_df = df.filter(df.Age > 30)
filtered_df.show()

# Perform aggregation
agg_df = df.groupBy("Name").avg("Age")
agg_df.show()

# Stop SparkSession when done
spark.stop()
```

Expected output:

```
from pyspark.sql import SparkSession

# Create a SparkSession
spark = SparkSession.builder \
    .appName("DataFrameExample") \
    .getOrCreate()

# Create a DataFrame from a list of tuples
data = [("John", 25), ("Alice", 30), ("Bob", 35)]
df = spark.createDataFrame(data, ["Name", "Age"])

# Show the DataFrame
df.show()

# Filter the DataFrame
filtered_df = df.filter(df.Age > 30)
filtered_df.show()

# Perform aggregation
agg_df = df.groupBy("Name").avg("Age")
agg_df.show()

# Stop SparkSession when done
spark.stop()
```

```
+-----+---+
| Name|Age|
+-----+---+
| John| 25|
|Alice| 30|
|  Bob| 35|
+-----+---+

+----+---+
|Name|Age|
+----+---+
| Bob| 35|
+----+---+

+-----+--------+
| Name|avg(Age)|
+-----+--------+
| John|    25.0|
|Alice|    30.0|
|  Bob|    35.0|
+-----+--------+
```

Give yourself a treat if you were able to run the code and get the expected output.

Thankyou for your time and patience!!

FIX

```
C:\WINDOWS\system32>set | findstr SPARK
PYSPARK_HOME=C:\Users\NikhilSharma\AppData\Local\Programs\Python\Python310\python.exe
PYSPARK_PYTHON=C:\Users\NikhilSharma\AppData\Local\Programs\Python\Python310
SPARK_HOME=C:\spark\spark-3.3.1-bin-hadoop2

C:\WINDOWS\system32>set | findstr HADOOP
HADOOP_HOME=C:\Hadoop

C:\WINDOWS\system32>set | findstr JAVA
JAVA_HOME=C:\Java\jdk

C:\WINDOWS\system32>"%JAVA_HOME%\bin\java" -version
java version "11.0.15" 2022-04-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.15+8-LTS-149)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.15+8-LTS-149, mixed mode)

C:\WINDOWS\system32>dir "%SPARK_HOME%\python\lib" | findstr py4j
10/15/2022  03:11 PM            42,404 py4j-0.10.9.5-src.zip

C:\WINDOWS\system32>dir "%PYSPARK_PYTHON%\Lib\site-packages" | findstr py4j
07/12/2024  05:08 PM    <DIR>          py4j
07/12/2024  05:08 PM    <DIR>          py4j-0.10.9.5.dist-info
```

1. First, let's check the Python environment that Jupyter is using:

   Open a new notebook and run this cell:

   ```
   import sys
   print(sys.executable)
   ```

   This will show you which Python installation Jupyter is using.
2. Now, let's check if PySpark is accessible in Jupyter:

```
import pyspark
print(pyspark.__version__)
```

Open a command prompt and run:

```
pip install --upgrade jupyter
pip install --upgrade pyspark==3.3.1
```

3. Configure Jupyter to use PySpark:
   You might need to create a Jupyter kernel specifically for PySpark. Create a file named `kernel.json` with this content:

```
{
 "argv": [
  "python",
  "-m",
  "ipykernel_launcher",
  "-f",
  "{connection_file}"
 ],
 "env": {
  "SPARK_HOME": "C:\\spark\\spark-3.3.1-bin-hadoop2",
  "PYSPARK_PYTHON":
"C:\\Users\\NikhilSharma\\AppData\\Local\\Programs\\Python\\Python310\\python.exe",
  "PYSPARK_DRIVER_PYTHON": "jupyter",
  "PYSPARK_DRIVER_PYTHON_OPTS": "notebook"
 },
 "display_name": "PySpark 3.3.1",
 "language": "python"
}
```

   Save this file in
   `C:\Users\YourUsername\AppData\Roaming\jupyter\kernels\pyspark3.3.1\kernel.json`

Create SparkSession

- What is a **SparkSession** - It is a representation or working instance of a Spark Application that is used to create and manage data processing in the system.
- What is **master("local[1]")** - defines whether to use local mode to run with only 1 execution thread.
- What is **appName("spark")** - Defines the name of the Spark Application.
- What is **getOrCreate()** - It's the method used to invoke or create a SparkSession.