**Complete Guide to Setting Up PySpark on Windows**

This guide walks you through installing and configuring PySpark on a Windows machine, addressing errors encountered during the process and ensuring compatibility with your ETL workflow. Follow each step carefully, and you’ll have a working PySpark environment.

**Step 1: Install Java 11**

**Why This Step Is Necessary**

PySpark requires a compatible Java Development Kit (JDK) to run. Initially, you had Java 21 installed, which caused the error:

* **java.lang.UnsupportedOperationException: getSubject is not supported**

This error occurred because Java 21 introduces stricter module and security restrictions incompatible with the Hadoop libraries used by PySpark (Spark 3.x). PySpark officially supports Java 8, 11, and 17. We chose Java 11 for its broad compatibility and stability with Spark.

**Instructions**

1. **Download Java 11:**
   * Visit [Eclipse Adoptium](https://adoptium.net/temurin/releases/).
   * Select **JDK 11** (e.g., version 11.0.15+10).
   * Download the Windows installer (.msi or .exe), such as jdk-11.0.15\_windows-x64\_bin.exe.
2. **Install Java 11:**
   * Run the installer and follow the prompts.
   * Note the installation path (e.g., C:\Program Files\Eclipse Adoptium\jdk-11.0.15.10-hotspot).
3. **Set JAVA\_HOME Environment Variable:**
   * Right-click **This PC** or **Computer** in File Explorer and select **Properties**.
   * Click **Advanced system settings** on the left.
   * In the System Properties window, click **Environment Variables**.
   * Under **System variables**, click **New** (or **Edit** if it exists):
     + **Variable name**: JAVA\_HOME
     + **Variable value**: C:\Program Files\Eclipse Adoptium\jdk-11.0.15.10-hotspot (adjust to your installation path).
   * Click **OK**.
4. **Update PATH Environment Variable:**
   * In the **System variables** section, find the Path variable and click **Edit**.
   * Click **New** and add %JAVA\_HOME%\bin.
   * Optionally, use **Move Up** to prioritize this version over others.
   * Click **OK** to save.
5. **Verify Java Installation:**
   * Open a new Command Prompt and run:

text

CollapseWrapCopy

java -version

* + Expected output:

text

CollapseWrapCopy

openjdk version "11.0.15" 2022-04-19

OpenJDK Runtime Environment Temurin-11.0.15+10 (build 11.0.15+10)

OpenJDK 64-Bit Server VM Temurin-11.0.15+10 (build 11.0.15+10, mixed mode)

**Details**

* **Version to Select:** JDK 11 (e.g., 11.0.15+10).
* **Link:** [Eclipse Adoptium](https://adoptium.net/temurin/releases/).
* **Error Resolved:** java.lang.UnsupportedOperationException: getSubject is not supported.

**Step 2: Set Up winutils.exe for Windows**

**Why This Step Is Necessary**

PySpark relies on Hadoop libraries, which require winutils.exe on Windows. Without it, you encountered:

* **Did not find winutils.exe: java.io.FileNotFoundException: HADOOP\_HOME and hadoop.home.dir are unset**

This error indicates that winutils.exe was missing, and the HADOOP\_HOME environment variable wasn’t set.

**Instructions**

1. **Download winutils.exe:**
   * Go to [this GitHub repository](https://github.com/cdarlint/winutils).
   * Navigate to hadoop-3.3.0/bin/winutils.exe and download it.
2. **Set Up Directory:**
   * Create a folder, e.g., C:\winutils.
   * Inside it, create a bin subfolder: C:\winutils\bin.
   * Place winutils.exe in C:\winutils\bin.
3. **Set HADOOP\_HOME Environment Variable:**
   * Right-click **This PC** or **Computer** in File Explorer and select **Properties**.
   * Click **Advanced system settings**.
   * Click **Environment Variables**.
   * Under **System variables**, click **New**:
     + **Variable name**: HADOOP\_HOME
     + **Variable value**: C:\winutils (use single backslashes).
   * Edit the Path variable, add %HADOOP\_HOME%\bin if not already present, and click **OK**.
4. **Verify winutils.exe:**
   * Open a Command Prompt and run:

text

CollapseWrapCopy

C:\winutils\bin\winutils.exe

* + If it runs (you may see a usage message), it’s working. If it fails, check permissions:
    - Right-click C:\winutils\bin\winutils.exe > **Properties** > **Security** > Grant your user **Full control**.

**Details**

* **Version to Select:** Hadoop 3.3.0.
* **Link:** [winutils.exe for Hadoop 3.3.0](https://github.com/cdarlint/winutils).
* **Error Resolved:** Did not find winutils.exe: java.io.FileNotFoundException.

**Step 3: Configure SparkSession to Use localhost**

**Why This Step Is Necessary**

Spark attempted to use your machine’s hostname, resulting in:

* **org.apache.spark.SparkException: Invalid Spark URL: spark://HeartbeatReceiver@Nicks\_Dell.myfiosgateway.com:50249**

This error arose due to Windows hostname resolution issues. Forcing Spark to use localhost bypasses this problem.

**Instructions**

1. **Update SparkSession Configuration:**
   * In your Python script, configure the SparkSession as follows:

python

CollapseWrapCopy

spark = SparkSession.builder \

.appName("Pandas to PySpark ETL") \

.master("local[\*]") \

.config("spark.driver.host", "localhost") \

.config("spark.driver.bindAddress", "127.0.0.1") \

.getOrCreate()

**Details**

* **Error Resolved:** org.apache.spark.SparkException: Invalid Spark URL.

**Step 4: Install Required Python Packages**

**Why This Step Is Necessary**

Your script requires specific Python libraries. You encountered:

* **ModuleNotFoundError: No module named 'distutils'**

This error occurred in Python 3.12 because distutils is no longer included by default, requiring setuptools.

**Instructions**

1. **Install Python Packages:**
   * Open a Command Prompt and run:

bash

CollapseWrapCopy

pip install pandas pyspark sqlalchemy pyodbc setuptools

**Details**

* **Links:**
  + [pandas](https://pandas.pydata.org/)
  + [pyspark](https://spark.apache.org/docs/latest/api/python/index.html)
  + [sqlalchemy](https://www.sqlalchemy.org/)
  + [pyodbc](https://pypi.org/project/pyodbc/)
  + [setuptools](https://pypi.org/project/setuptools/)
* **Error Resolved:** ModuleNotFoundError: No module named 'distutils'.

**Step 5: Restart Terminal and PyCharm**

**Why This Step Is Necessary**

Environment variable changes (e.g., JAVA\_HOME, HADOOP\_HOME) don’t take effect until the terminal and applications like PyCharm are restarted.

**Instructions**

1. **Close and Reopen Terminal:**
   * Close all Command Prompt or terminal windows.
   * Open a new one.
2. **Restart PyCharm:**
   * Close PyCharm completely.
   * Reopen it and load your project.

**Details**

* Ensures the updated environment variables are applied.

**Step 6: Update Hosts File (Optional)**

**Why This Step Is Necessary**

If hostname issues persist, manually mapping your hostname to 127.0.0.1 can resolve:

* **org.apache.spark.SparkException: Invalid Spark URL**

**Instructions**

1. **Edit Hosts File:**
   * Open C:\Windows\System32\drivers\etc\hosts in a text editor (run as administrator).
   * Add:

text

CollapseWrapCopy

127.0.0.1 Nicks\_Dell.myfiosgateway.com

* + Save and close.

**Details**

* **Error Resolved:** org.apache.spark.SparkException: Invalid Spark URL (if still occurring).

**Step 7: Test PySpark Setup**

**Why This Step Is Necessary**

To confirm PySpark is working before running your ETL script.

**Instructions**

1. **Run a Test Script:**
   * Use this code:

python

CollapseWrapCopy

import os

os.environ['HADOOP\_HOME'] = 'C:\\winutils'

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Test").getOrCreate()

df = spark.createDataFrame([(1, "a"), (2, "b")], ["id", "value"])

df.show()

spark.stop()

* + If it runs and displays the DataFrame, your setup is successful.

**Step 8: Run Your ETL Script**

**Why This Step Is Necessary**

This is your goal: running an ETL script with PySpark. You mentioned issues converting back to a Pandas DataFrame, but the setup works for PySpark operations.

**Instructions**

1. **Update Your Script:**
   * Use this template, adjusting paths and credentials:

python

CollapseWrapCopy

import os

os.environ['HADOOP\_HOME'] = 'C:\\winutils'

import pandas as pd

from pyspark.sql import SparkSession

from sqlalchemy import create\_engine

*# Initialize SparkSession*

spark = SparkSession.builder \

.appName("Pandas to PySpark ETL") \

.master("local[\*]") \

.config("spark.driver.host", "localhost") \

.config("spark.driver.bindAddress", "127.0.0.1") \

.getOrCreate()

*# Load Pandas DataFrame*

pandas\_df = pd.read\_excel('C:\\path\\to\\your\\file.xlsx')

*# Convert to PySpark DataFrame*

spark\_df = spark.createDataFrame(pandas\_df)

*# Run SQL query*

spark\_df.createOrReplaceTempView("my\_table")

result\_df = spark.sql("SELECT \* FROM my\_table")

*# (Optional) Convert back to Pandas and export*

pandas\_result\_df = result\_df.toPandas()

connection\_string = 'mssql+pyodbc://username:password@server/database?driver=ODBC+Driver+17+for+SQL+Server'

engine = create\_engine(connection\_string)

pandas\_result\_df.to\_sql('table\_name', con=engine, if\_exists='replace', index=False)

*# Stop SparkSession*

spark.stop()

**Details**

* **Error Resolved:** java.net.SocketException: Connection reset by peer (related to hostname, fixed with localhost).

**Additional Notes**

* **Performance:** Converting between Pandas and PySpark adds overhead. For small datasets, Pandas alone may suffice; PySpark excels with large datasets.
* **Java Updates:** If you update Java later, adjust JAVA\_HOME and Path accordingly.
* **Troubleshooting:** If new errors arise, note the exact messages and share them for further assistance.

This documentation should enable you to set up PySpark on your other Windows virtual machines successfully. Let me know if you need help with specific steps or new issues!