Installation of Apache Hadoop on Windows

Step 1: Download Hadoop, Java and winutils from below links.

- We're going to install Hadoop version 3.1.0
    - Hadoop 3.1.0
        - https://archive.apache.org/dist/hadoop/common/hadoop-3.1.0/
    - Java
        - https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
    - Winutils
        - https://github.com/namra98/Hadoop-3.1.0-winutils
- Extract all three files (Hadoop will take a while).
- Run the Java installer but change the destination folder from the default "C:\Program Files\Java\" to just "C:\Java".
- Make a directory "C:\hadoop".
- Move all contents of Hadoop-3.1.0 to C:\Hadoop

Step 2: Setup Environment variables

- Go to Control Panel > System and Security > System > Advanced System Settings > Environment Variables
- Add new System variables (bottom box) called:
    - JAVA_HOME --> C:\Java
    - HADOOP_HOME --> C:\hadoop
- Edit **Path** and add the following:
    - C:\Java\bin
    - C:\hadoop\bin
    - C:\hadoop\sbin
- Check in by opening CMD (Win + R -> cmd):
    - java -version
    - hdfs – version

Step 3: Setup Hadoop Configurations

- Go to C:\hadoop\etc\hadoop and edit\create core-site.xml

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

- In the same directory, edit (or create) mapred-site.xml with the following contents:

```
<configuration>
```

```
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- Next, edit (or create) hdfs-site.xml:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///C:/hadoop/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///C:/hadoop/datanode</value>
  </property>
</configuration>
```

- Finally, edit yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

- The last thing we need to do is create the directories that we referenced in hdfs-site.xml (DIY).

Step 4: Patch Hadoop

- Hadoop was mainly created for Unix, so we need to patch it to make it work on windows.
- Just copy bin folder from winutils and paste it in C:\hadoop.
- If it asks for overwrite then allow it.
- copy **hadoop-yarn-server-timelineservice-3.0.3** from C:\hadoop\share\hadoop\yarn\timelineservice to C: \hadoop\share\hadoop\yarn (the parent directory).

Step 5: Run HDFS

- Open CMD and run **hdfs namenode -format**
- Finally, you can boot HDFS by running start-dfs.cmd and start-yarn.cmd in cmd. (It will open 4 cmd windows).
- You can monitor these windows by typing "**jps**" in cmd.
- You can see list of all applications by opening http://localhost:8088

Congratulations, if you are here without any error (or by solving many errors) you've successfully installed Hadoop cluster in your windows machine.

References :

- https://dev.to/awwsmm/installing-and-running-hadoop-and-spark-on-windows-33kc
- https://github.com/MuhammadBilalYar/Hadoop-On-Window/wiki/How-to-Run-Hadoop-wordcount-MapReduce-Example-on-Windows-10

Running Your First MapReduce in Hadoop & Java:

Step 1: Download Jar and Data

- Download it from: `https://github.com/namra98/Hadoop_MapReduce`

Step 2: Start Hadoop Cluster

- `start-all.cmd`

Step 3: Create an input directory in HDFS.

- `hadoop fs -mkdir /input_dir`

Step 4: Copy the input text file named input_file.txt in the input directory (input_dir)of HDFS.

- `hadoop fs -put C:/input_file.txt /input_dir`

Step 5: Verify input_file.txt available in HDFS input directory (input_dir).

- `hadoop fs -ls /input_dir/`

Step 6: Verify content of the copied file.

- `Verify content of the copied file.`

Step 7: Run MapReduceClient.jar and provide input and out directories.

- `hadoop jar MapReduceClient.jar wordcount /input_dir /output_dir`

Step 8: Verify content for generated output file.

- `hadoop dfs -cat /output_dir/*`

Step 8: Save content of generated output file to local disk.

- `hadoop dfs -get /output_dir/* localfolder`

Running your Program with python:

- Start HDFS server : `start-all.cmd`
- Format Namenode : `hdfs namenode -format`
- Copy data file to HDFS : `hdfs dfs -put data.txt /`
- List files (same as unix) : `hdfs dfs -ls /`
- Run MapReduce Job :

  -
    ```
    hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.1.0.jar -file path\to\mapper.py -mapper "python mapper.py" -file path\to\reducer.py -reducer "python reducer.py" -input text.txt -output out
    ```
- It will save file in HDFS and you must bring it back via:
  - `hdfs dfs -get /user/hadoop/file localfile`