Steps to execute WordCount program on Hadoop using MapR Sandbox:

1. Upload input data (input.txt) to Hadoop
   1. Create a folder and go to the folder on Local File System

|  |
| --- |
| ls -lrth |
| mkdir -p /home/mapr/wordcount |
| ls -lrth |
| cd $HOME/wordcount |

Text

Description automatically generated

* 1. Create a input.txt file for WordCount program at Local File System.

|  |
| --- |
| vi input.txt |

* 1. Copy following contents into the files

|  |
| --- |
| Hadoop is the Elephant King!  A yellow and elegant thing.  He never forgets  Useful data, or lets  An extraneous element cling! |

|  |
| --- |
| Press i - It should show insert |
| Right Click copy paste the above file content. |
| Press Esc key |
| :wq! |
| Press Enter key |

Or You can use tool like Winscp / File Zilla to upload input.txt file

* 1. Verify content of the file

|  |
| --- |
| cat $HOME/wordcount/input.txt |

* 1. Create a directory in Hadoop file system and Check directory created in Hadoop File System or not.

|  |
| --- |
| hadoop fs -mkdir -p wordcount/input |
| hadoop fs -ls -R wordcount |

Text

Description automatically generated

* 1. Copy file from Local File System to Hadoop File System.

|  |
| --- |
| hadoop fs -put $HOME/wordcount/input.txt wordcount/input/ |

* 1. Check file added to Hadoop file system or not.

|  |
| --- |
| hadoop fs -ls -R wordcount |
| hadoop fs -cat wordcount/input/input.txt |

Text

Description automatically generated

1. Executing Map Reduce Python Script using Hadoop
   1. Create a directory in the Local File System

|  |
| --- |
| mkdir -p /home/mapr/wordcount/python |

* 1. Go to the directory in the Local File System.

|  |
| --- |
| cd /home/mapr/wordcount/python |

* 1. Upload the python 'mapper.py' , 'reducer.py' files using WinScp / File Zilla etc tools.

Graphical user interface, application

Description automatically generated

* 1. Run Locally Code using Python.

|  |
| --- |
| cat $HOME/wordcount/input.txt | python $HOME/wordcount/python/mapper.py | sort -k1,1 | python $HOME/wordcount/python/reducer.py |

* 1. To give permissions to python scripts.

|  |
| --- |
| chmod 777 mapper.py reducer.py |

* 1. Convert Python script to Unix compatible files.

|  |
| --- |
| dos2unix \*.py |

* 1. Make sure and Check if python script files has following lines on the top of the python script code as commented.

#!/usr/bin/env python

|  |
| --- |
| head -10 mapper.py |
| head -10 reducer.py |

Graphical user interface, text, application

Description automatically generated

* 1. In case output folder exists already then make sure to delete it.

|  |
| --- |
| hadoop fs -rm -r wordcount/pythonoutput |

* 1. Execute the Map Reduce job using Hadoop / Python.

|  |
| --- |
| hadoop jar /opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/tools/lib/hadoop-streaming-2.7.0-mapr-1808.jar -input wordcount/input/input.txt -output wordcount/pythonoutput -mapper /home/mapr/wordcount/python/mapper.py -reducer /home/mapr/wordcount/python/reducer.py |

* 1. Check the output directory into the Hadoop File System

|  |
| --- |
| hadoop fs -ls -R wordcount/pythonoutput/ |

* 1. Check the output file content from the Hadoop File System

|  |
| --- |
| hadoop fs -cat wordcount/pythonoutput/part-00000 |

Text

Description automatically generated

Text, letter

Description automatically generated

1. Executing WordCount Java Program using Hadoop.
   1. Creating a WordCount.jar file on Local File System
      1. Check Hadoop version

|  |
| --- |
| hadoop version |

* + 1. Create 2 variables

Using output of the hadoop version findout hadoop path

For e.g., /opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/hadoop-common-2.7.0-mapr-1808.jar

Then Hadoop Home path should be /opt/mapr/hadoop/hadoop-2.7.0/

|  |
| --- |
| export HADOOP\_HOME=/opt/mapr/hadoop/hadoop-2.7.0/ |
| export HADOOP\_CLASSPATH=$(find $HADOOP\_HOME -name '\*.jar' | xargs echo | tr ' ' ':') |

* + 1. Create a folder on Local File System to save WordCount Java programs class files.

|  |
| --- |
| mkdir -p $HOME/wordcount/wordcount\_class |

* + 1. Upload WordCount.java file at location $HOME/wordcount using WinScp/FileZilla tool.

Graphical user interface, text, application

Description automatically generated

* + 1. Check WordCount.java program file is present at local file system

|  |
| --- |
| ls $HOME/wordcount/WordCount.java |

* + 1. Command to create compile Java program WordClass.java this will create java class files.

|  |
| --- |
| javac -classpath $HADOOP\_CLASSPATH -d "$HOME/wordcount/wordcount\_class" "$HOME/wordcount/WordCount.java" |

* + 1. Check directory and class files should be present at Local File System.

|  |
| --- |
| ls -lrth $HOME/wordcount/wordcount\_class |

* + 1. Create a WordCount.jar file that can be run on Hadoop.

|  |
| --- |
| jar -cvf WordCount.jar -C wordcount\_class/ . |

* + 1. Check WordCount.jar created or not

|  |
| --- |
| ls -lrth $HOME/wordcount/ |

Text

Description automatically generated

* 1. Execute WordCount.jar using Hadoop

|  |
| --- |
| hadoop jar WordCount.jar WordCount wordcount/input wordcount/output |

* 1. Check Hadoop File System output folder

|  |
| --- |
| hadoop fs -ls -R wordcount/output |

* 1. View the WordCount program output from Hadoop File System.

|  |
| --- |
| hadoop fs -cat wordcount/output/part-r-00000 |

Text

Description automatically generated

Text, letter

Description automatically generated

Graphical user interface, text, application

Description automatically generated