

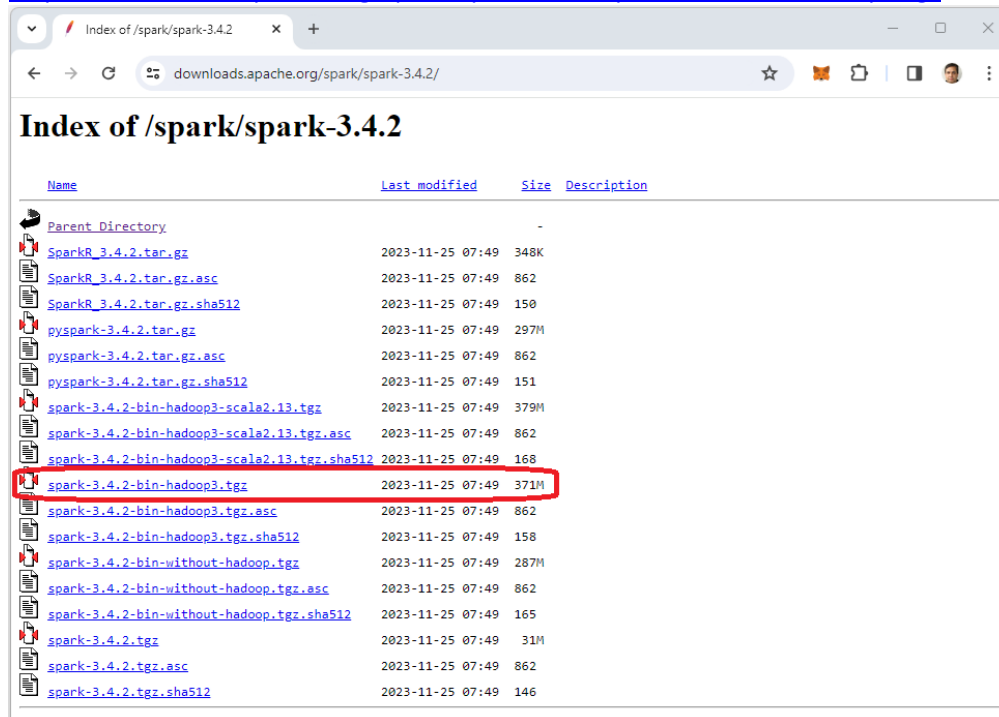
Tutorial 4 (Part II)

Apache Spark in Jupyter

Install and test Apache Spark in VirtualBox VM

- 1) Download Apache Spark 3.4.2 prebuilt for Hadoop3 from the link below in the **Downloads** folder by typing the web address in the Firefox browser

<https://downloads.apache.org/spark/spark-3.4.2/spark-3.4.2-bin-hadoop3.tgz>



After download completion, unzip it using the following command.

Check the download folder is present or not using (**\$ls**) command. Do not write \$ sign as it showed prompt in Ubuntu shell

```
$cd Downloads
```

```
$sudo tar -xvf spark-3.4.2-bin-hadoop3.tgz
```

OR use command line to download if you have any difficulty in downloading with the Mozilla Firefox browser. The below highlighted commands do not need to execute if you have already downloaded Apache Spark. If you have difficulty in downloading using browser, you can use terminal or shell to download Apache Spark.

Use the following command

```
$cd /home/hduser/Downloads
```

```
$wget https://downloads.apache.org/spark/spark-3.4.2/spark-3.4.2-bin-hadoop3.tgz
```

```
$ls -l
```

You can check the zip folder after this download and now you can unzip the folder as mentioned below

```
$sudo tar -xvf spark-3.4.2-bin-hadoop3.tgz
```

- 2) Install Apache Spark under **/usr/local** by running the commands

```
$sudo mv ./spark-3.4.2-bin-hadoop3 /usr/local
```

```
$cd /usr/local
```

- 3) Create a symbolic link called **spark** to the spark-3.4.2-bin-hadoop3.2:

```
$sudo ln -sf ./spark-3.4.2-bin-hadoop3 spark
```

- 4) Change the ownership of the files in the **spark** directory so that the group is assigned to **hadoop** and the owner is **hduser**:

```
$sudo chown -R hduser:hadoopgroup spark*
```

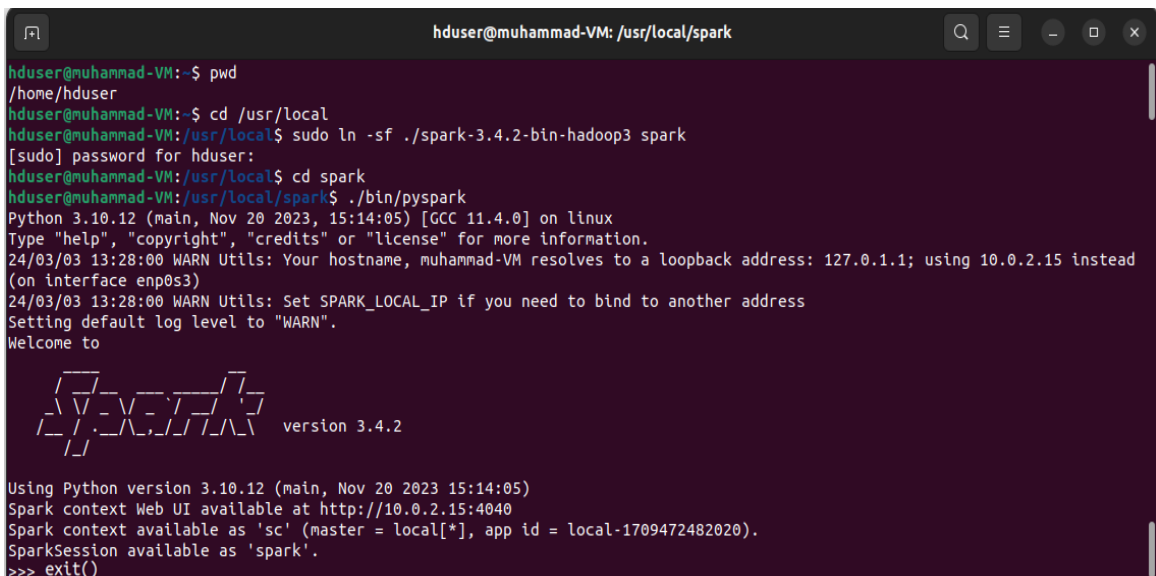
- 5) Test the installation at the path `‘/usr/local/spark’`

```
$cd /usr/local/spark
```

and then

```
$./bin/pyspark
```

Did you get the Python shell as shown in the screenshot below? OK. If you did not get the python shell as mentioned below, then check the previous steps again.



```
hduser@muhammad-VM: /usr/local/spark
hduser@muhammad-VM:~$ pwd
/home/hduser
hduser@muhammad-VM:~$ cd /usr/local
hduser@muhammad-VM: /usr/local$ sudo ln -sf ./spark-3.4.2-bin-hadoop3 spark
[sudo] password for hduser:
hduser@muhammad-VM: /usr/local$ cd spark
hduser@muhammad-VM: /usr/local/spark$ ./bin/pyspark
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
24/03/03 13:28:00 WARN Utils: Your hostname, muhammad-VM resolves to a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface enp0s3)
24/03/03 13:28:00 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | || |___) |
| |  | || |___) |
| |  | || |___) |
|_|  |_| \____/

version 3.4.2

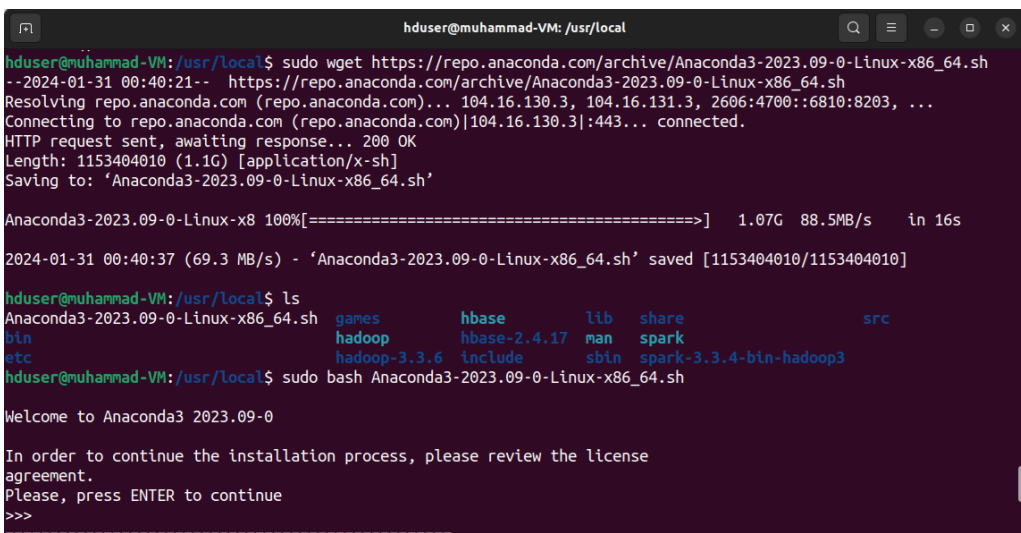
Using Python version 3.10.12 (main, Nov 20 2023 15:14:05)
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1709472482020).
SparkSession available as 'spark'.
>>> exit()
```

- 6) Install Anaconda using the following command:

```
$cd /usr/local
```

```
$sudo wget https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
```

```
$sudo bash Anaconda3-2023.09-0-Linux-x86_64.sh
```



```
hduser@muhammad-VM: /usr/local
hduser@muhammad-VM: /usr/local$ sudo wget https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
--2024-01-31 00:40:21-- https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.130.3, 104.16.131.3, 2606:4700::6810:8203, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.130.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1153404010 (1.1G) [application/x-sh]
Saving to: 'Anaconda3-2023.09-0-Linux-x86_64.sh'

Anaconda3-2023.09-0-Linux-x86 100%[=====] 1.07G 88.5MB/s in 16s

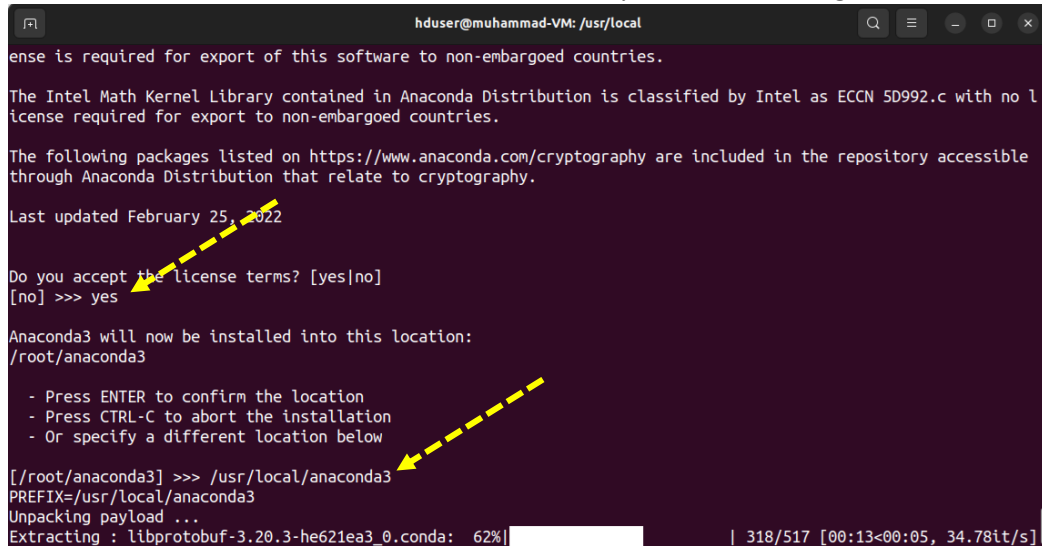
2024-01-31 00:40:37 (69.3 MB/s) - 'Anaconda3-2023.09-0-Linux-x86_64.sh' saved [1153404010/1153404010]

hduser@muhammad-VM: /usr/local$ ls
Anaconda3-2023.09-0-Linux-x86_64.sh  games      hbase      lib  share      src
bin                                hadoop     hbase-2.4.17  man  spark
etc                                hadoop-3.3.6  include  sbin  spark-3.3.4-bin-hadoop3
hduser@muhammad-VM: /usr/local$ sudo bash Anaconda3-2023.09-0-Linux-x86_64.sh

Welcome to Anaconda3 2023.09-0

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

Press the Enter Key and use arrow key to check the terms and conditions. Use arrow key to move to the end of this document. Then accept the license conditions as “yes” and choose option for **/usr/local/anaconda3** as the install directory and the following screenshot will appear as



```

hduser@muhammad-VM: /usr/local
ense is required for export of this software to non-embargoed countries.

The Intel Math Kernel Library contained in Anaconda Distribution is classified by Intel as ECCN 5D992.c with no l
icense required for export to non-embargoed countries.

The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible
through Anaconda Distribution that relate to cryptography.

Last updated February 25, 2022

Do you accept the license terms? [yes/no]
[no] >>> yes

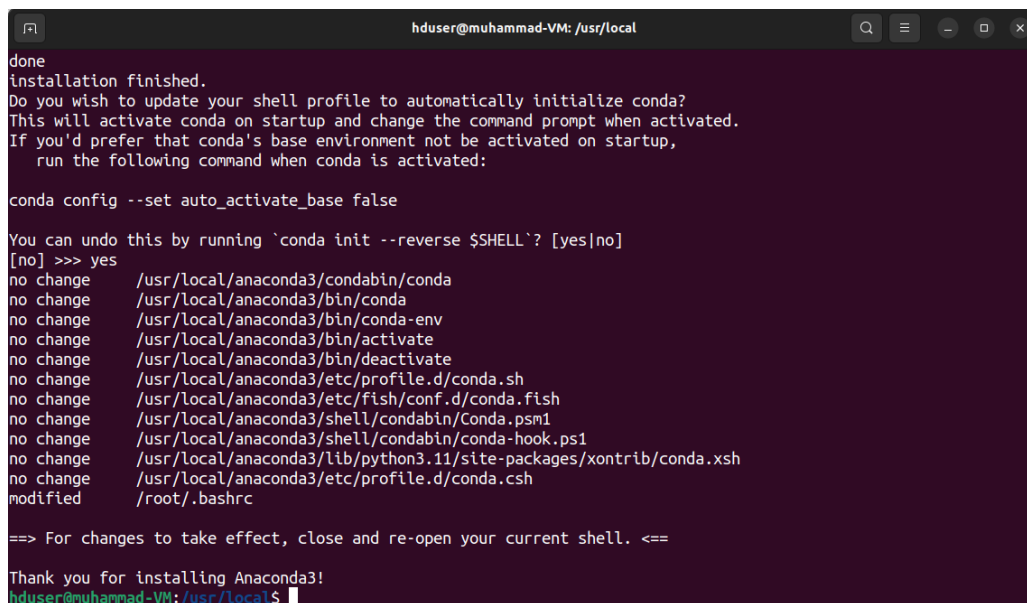
Anaconda3 will now be installed into this location:
/root/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/root/anaconda3] >>> /usr/local/anaconda3
PREFIX=/usr/local/anaconda3
Unpacking payload ...
Extracting : libprotobuf-3.20.3-he621ea3_0.conda: 62%|          | 318/517 [00:13<00:05, 34.78it/s]

```

When the installation of anaconda3 will be completed, you will get the option for yes/ no for running anaconda.



```

hduser@muhammad-VM: /usr/local
done
installation finished.
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated.
If you'd prefer that conda's base environment not be activated on startup,
run the following command when conda is activated:

conda config --set auto_activate_base false

You can undo this by running `conda init --reverse $SHELL'? [yes/no]
[no] >>> yes
no change /usr/local/anaconda3/condabin/conda
no change /usr/local/anaconda3/bin/conda
no change /usr/local/anaconda3/bin/conda-env
no change /usr/local/anaconda3/bin/activate
no change /usr/local/anaconda3/bin/deactivate
no change /usr/local/anaconda3/etc/profile.d/conda.sh
no change /usr/local/anaconda3/etc/fish/conf.d/conda.fish
no change /usr/local/anaconda3/shell/condabin/Conda.psm1
no change /usr/local/anaconda3/shell/condabin/conda-hook.ps1
no change /usr/local/anaconda3/lib/python3.11/site-packages/xontrib/conda.xsh
no change /usr/local/anaconda3/etc/profile.d/conda.csh
modified /root/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

Thank you for installing Anaconda3!
hduser@muhammad-VM: /usr/local$

```

7) Open a new terminal and add Spark environment variables to the **.bashrc** file from **/home/hduser**:

```
$cd /home/hduser
```

```
$nano ./bashrc
```

Add the following lines in the **./bashrc** file using **nano** editor at the end of the file. Save the file using nano editor.

```

# Spark configuration
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
export PYSPARK_PYTHON=/usr/local/anaconda3/bin/python3
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_PYTHON=python3
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"

```

```

GNU nano 6.2 ~/.bashrc
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# Hadoop config
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

# Native path
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_HOME}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

# Java path
export JAVA_HOME="/usr"

# OS path
export PATH=$PATH:$HADOOP_HOME/bin:$JAVA_PATH/bin:$HADOOP_HOME/sbin

# Spark configuration
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
export PYSARK_PYTHON=/usr/local/anaconda3/bin/python3
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_PYTHON=python3
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^G Location  M-U Undo
^X Exit      ^R Read File ^_ Replace   ^J Paste     ^I Justify  ^_ Go To Line M-E Redo

```

8) Save the above commands and source your **bashrc** with the command to update:

```
$source ~/.bashrc
```

9) Test to see if Spark launches with the command:

```
$pyspark
```

NOTE: A web page is launched instead of the ripple launched earlier since we have opted (in the bashrc file) to launch a Jupyter notebook and we will write our python code there.

If you see some errors like as mentioned on the below screenshot, then install jupyter installer

```

hduser@muhammad-Vm:~$ cd /home/hduser
hduser@muhammad-Vm:~$ nano ~/.bashrc
hduser@muhammad-Vm:~$ source ~/.bashrc
hduser@muhammad-Vm:~$ pyspark
env: 'jupyter': No such file or directory
hduser@muhammad-Vm:~$ sudo apt install jupyter

```

Use the command to launch **pyspark** and you should see a terminal in the browser will be opened. If you found an error like, env: 'jupyter': No such file or directory. Then install the jupyter notebook by using the command

```
$sudo apt install jupyter
```

Write 'y' when asked for the installation.

10) After installation of jupyter, write again **pyspark**

```
$pyspark
```

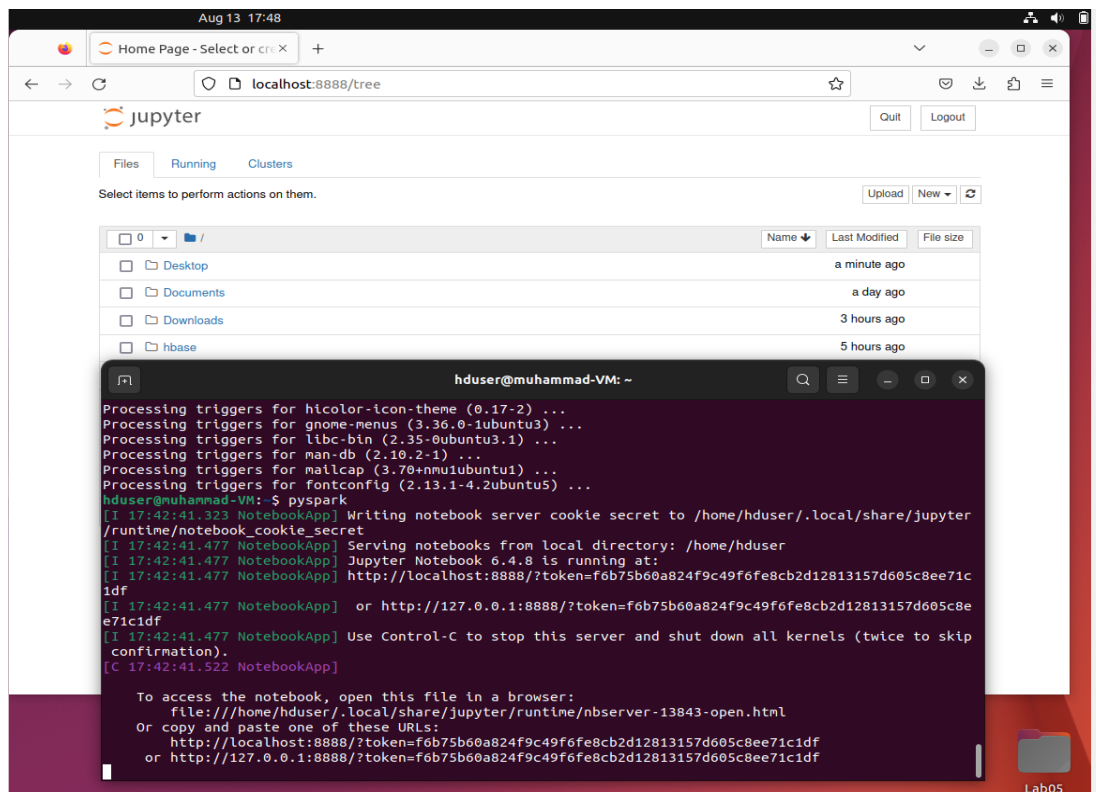
The jupyter notebook started automatically using google chrome browser (Installation of Google chrome is provided in Appendix A) or you can copy the link address from the terminal and paste in the Mozilla Firefox.

```

hduser@muhammad-VM: ~
Setting up python3-ipywidgets (6.0.0-9) ...
Setting up python3-dev (3.10.6-1-22.04) ...
Setting up python3-pythran (0.10.0+ds2-1) ...
Setting up python3-scipy (1.8.0-1exp2ubuntu1) ...
Setting up python3-fonttools (4.29.1-2build1) ...
Setting up python3-ufolib2 (0.13.1+dfsg1-1) ...
Setting up python3-matplotlib (3.5.1-2build1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
hduser@muhammad-VM:~$ pyspark
[I 17:42:41.323 NotebookApp] Writing notebook server cookie secret to /home/hduser/.local/share/jupyter/runtime/notebook_cookie_secret
[I 17:42:41.477 NotebookApp] Serving notebooks from local directory: /home/hduser
[I 17:42:41.477 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 17:42:41.477 NotebookApp] http://localhost:8888/?token=f6b75b60a824f9c49f6fe8cb2d12813157d605c8ee71c1df
[I 17:42:41.477 NotebookApp] or http://127.0.0.1:8888/?token=f6b75b60a824f9c49f6fe8cb2d12813157d605c8ee71c1df
[I 17:42:41.477 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 17:42:41.522 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/hduser/.local/share/jupyter/runtime/nbserver-13843-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=f6b75b60a824f9c49f6fe8cb2d12813157d605c8ee71c1df
or http://127.0.0.1:8888/?token=f6b75b60a824f9c49f6fe8cb2d12813157d605c8ee71c1df

```



11) Open a new terminal and create a working directory (**Lab04**) for the Spark.

```

$cd /home/hduser/Desktop
$pwd
$mkdir Lab04
$cd Lab04
$pwd

```

12) Now we copy some text, e.g. song lyrics from: <http://www.songlyrics.com/> into a file **sample_lyric.txt** or Download the file **pg30123.txt** from Moodle. Store this file in Lab04 folder.

13) Start hadoop using the commands (**\$start-dfs.sh** and **\$start-yarn.sh**)

```

hduser@muhammad-VM: ~/Desktop/Lab04
hduser@muhammad-VM:~$ pwd
/home/hduser
hduser@muhammad-VM:~$ cd Desktop
hduser@muhammad-VM:~/Desktop$ mkdir Lab04
hduser@muhammad-VM:~/Desktop$ cd Lab04
hduser@muhammad-VM:~/Desktop/Lab04$ pwd
/home/hduser/Desktop/Lab04
hduser@muhammad-VM:~/Desktop/Lab04$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [muhammad-VM]
hduser@muhammad-VM:~/Desktop/Lab04$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@muhammad-VM:~/Desktop/Lab04$ jps
4259 ResourceManager
3684 NameNode
3831 DataNode
4057 SecondaryNameNode
4713 Jps
4395 NodeManager
hduser@muhammad-VM:~/Desktop/Lab04$

```

- 14) Download the zip file from Moodle, Tutorial_4_Spark_WordCount_Example.zip, unzip it and copy both files into Lab04 folder on Desktop using the commands.

```

$cd Downloads
$ls
$unzip ./Tutorial_4_Spark_WordCount_Example.zip -d
/home/hduser/Desktop/Lab04

```

```

hduser@muhammad-VM: ~/Desktop/Lab04
hduser@muhammad-VM:~/Downloads$ ls Tutorial_4*
hduser@muhammad-VM:~/Downloads$ unzip ./Tutorial_4_Spark_WordCount_Example.zip -d /home/hduser/Desktop/Lab04
Archive:  ./Tutorial_4_Spark_WordCount_Example.zip
  inflating: /home/hduser/Desktop/Lab04/pg30123.txt
  inflating: /home/hduser/Desktop/Lab04/Tutorial_4_Spark_WordCount.ipynb
hduser@muhammad-VM:~/Downloads$ cd /home/hduser/Desktop/Lab04
hduser@muhammad-VM:~/Desktop/Lab04$ ls
pg30123.txt  Tutorial_4_Spark_WordCount.ipynb
hduser@muhammad-VM:~/Desktop/Lab04$ jps
4259 ResourceManager
3684 NameNode
3831 DataNode
15879 Jps
4057 SecondaryNameNode
4395 NodeManager
hduser@muhammad-VM:~/Desktop/Lab04$ hadoop fs -put ./pg30123.txt /user1
hduser@muhammad-VM:~/Desktop/Lab04$ hadoop fs -ls /user1/pg30123.txt
-rw-r--r--  1 hduser supergroup    65167 2024-03-03 12:03 /user1/pg30123.txt
hduser@muhammad-VM:~/Desktop/Lab04$ pyspark
[I 12:11:23.296 NotebookApp] Serving notebooks from local directory: /home/hduser/Desktop/Lab04
[I 12:11:23.296 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 12:11:23.296 NotebookApp] http://localhost:8888/?token=9bfe4a51917fdaf02cb7c2841079dce9be3c452b40731362
[I 12:11:23.296 NotebookApp] or http://127.0.0.1:8888/?token=9bfe4a51917fdaf02cb7c2841079dce9be3c452b40731362
[I 12:11:23.296 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 12:11:23.317 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/hduser/.local/share/jupyter/runtime/nbserver-16196-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=9bfe4a51917fdaf02cb7c2841079dce9be3c452b40731362
or http://127.0.0.1:8888/?token=9bfe4a51917fdaf02cb7c2841079dce9be3c452b40731362

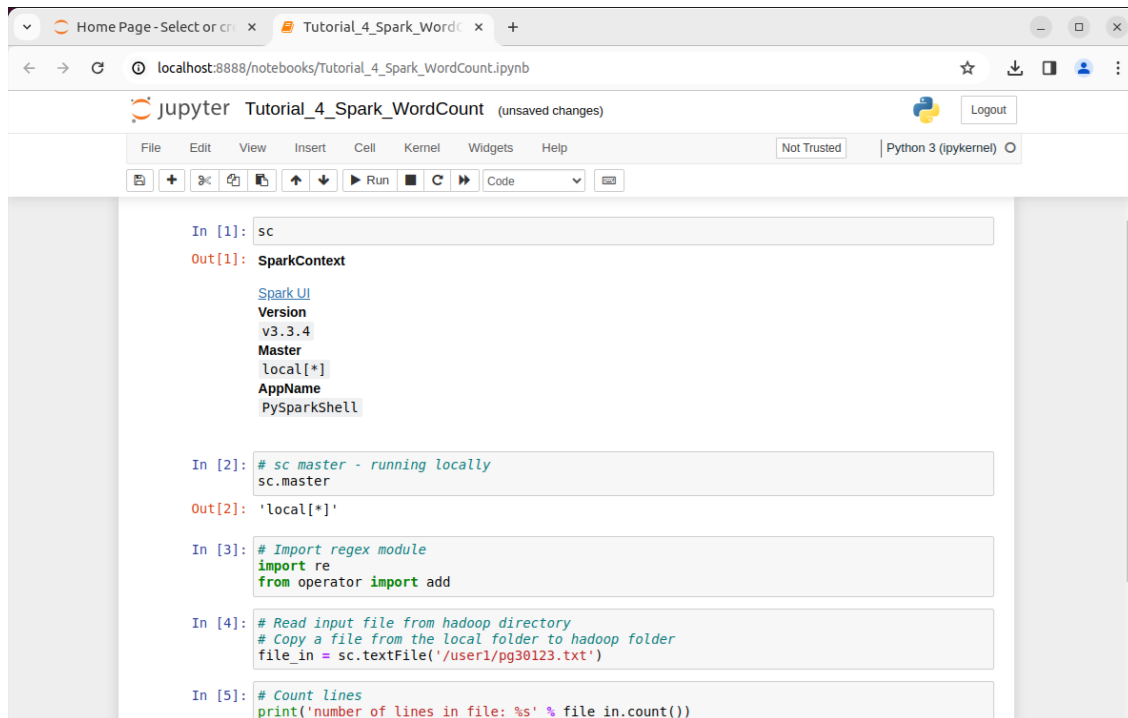
```

Move “pg30123.txt” file from Lab04 folder on your local VM into the folder on hadoop distributed file system (hdfs) before launching the wordcount code in pyspark as shown in the above screenshot.

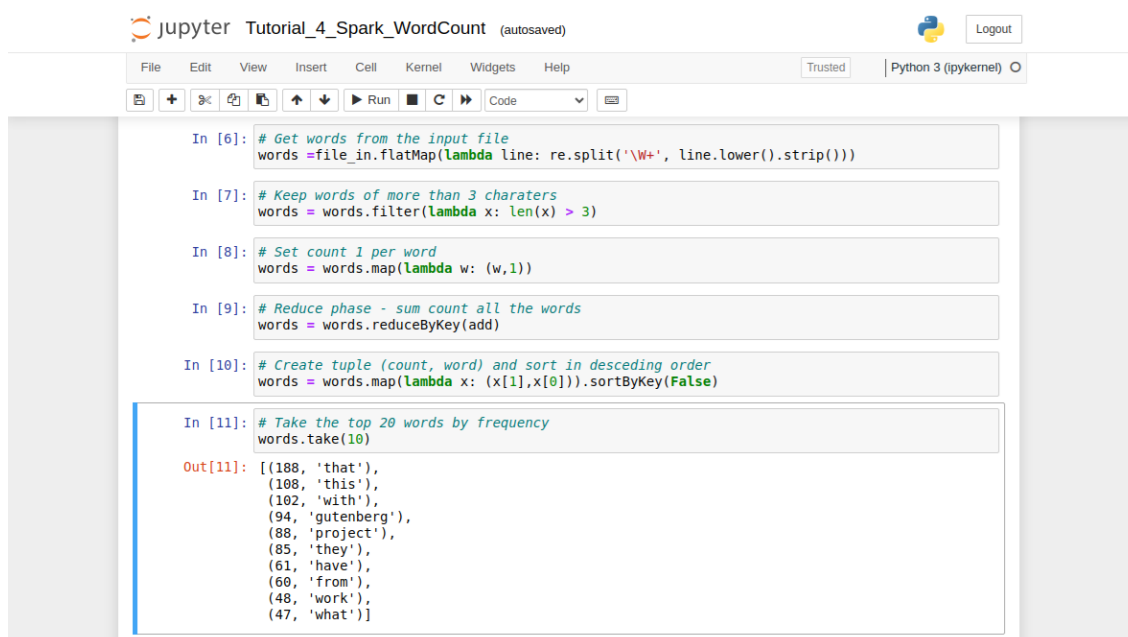
- 15) Now again launch Pyspark or you can move to the folder in Lab04 in the already opened jupyter notebook.



16) In the browser Jupyter interface navigate and open the Tutorial_4_Spark_WordCount.ipynb



17) Test the code and analyse what it does. Make sure the input file should be present on the hadoop distributed file system.



18) Practice Tutorial exercises provided for further exploration Apache spark on Moodle.

Appendix A

If you would like to install google chrome, Follow the following steps as mentioned below

```
$sudo apt update
$ wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
$sudo dpkg -i google-chrome-stable_current_amd64.deb
$google-chrome
```

Add to your favorites when Google Chrome launches so that you won't need to use the terminal to launch it again.

References:

- <https://spark.apache.org/examples.html>
- <https://www.cloudduggu.com/spark/>