Toronto, Canada

A Report on

Moule 1 Lab 0 : Grab Azure VM/Create a Pyspark Cluster

Data Management and Big Data

(ALY6010)

Guided by:

Dr. Mohammad Shafiqul Islam

Submitted by:

Pratikkumar Indravadan Malaviya

NUID : 002963548

Date of submission : 11th November'2022

Introduction

The main aim of this lab is to get comfortable with setting up Single Node Cluster and PySpark on hardware in order to install Java, Python, and Jupyter notebook. We will build up a single node cluster in an Ubuntu virtual machine to run PySpark (VM). According to your preferences and operating system, we can build up an Ubuntu virtual machine. A single Apache Spark driver and no Spark workers make up a single node cluster. All Spark jobs and data sources, including Delta Lake, can be run on a single node cluster. Spark jobs must be executed by at least one Spark worker in a Standard cluster.

Installation

1. The first step is to enable specific functionality from the Windows Features menu, such as the Virtual machine platform and Windows subsystem for Linux.

2. Second, we'll use the browser to download the most recent version of Ubuntu. Once Ubuntu has been successfully downloaded, we'll open a terminal session, wait for the installation to be finished, and then, if requested, choose a new Linux username and password. The new Linux command line is currently active.

3. Executing the update and upgrade command, which only downloads the required packages, comes after installation.
   (“sudo apt update && upgrade”)
   The "sudo apt-get upgrade" command downloads and install the most recent versions of all out-of-date system dependencies and packages.

```
pratik@LAPTOP-88JUMD7U:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
pratik@LAPTOP-88JUMD7U:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Figure 1 Update and Upgrade package

4. The most recent version of Java will be downloaded and installed in the following step. The Pyspark cluster's necessary Java JDK will be installed via the command line.
   ($sudo apt-get install openjdk-11-jre)
   ($sudo apt-get install openjdk-11-jdk)

After that, we can use the command "$ java -version" to view the version details.

```
pratik@LAPTOP-88JUMD7U:~$ sudo apt-get install openjdk-11-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-11-jre is already the newest version (11.0.17+8-1ubuntu2~20.04).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pratik@LAPTOP-88JUMD7U:~$ sudo apt-get install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.17+8-1ubuntu2~20.04).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Figure 2 Installing JDK and JRE

5. Moving on to the second installation phase, we will now install Python 3. simply because PySpark is the Python API for Apache Spark, a free and open-source platform for distributed computing that includes a number of tools for processing massive amounts of data quickly.
   ($sudo apt install python3 python3-pip ipython3)
   ($sudo apt install python3-pip)

```
pratik@LAPTOP-88JUMD7U:~$ sudo apt install python3 python3-pip ipython3
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
ipython3 is already the newest version (7.13.0-1).
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pratik@LAPTOP-88JUMD7U:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pratik@LAPTOP-88JUMD7U:~$ pip3 install jupyter py4j pyspark
Requirement already satisfied: jupyter in ./.local/lib/python3.8/site-packages (1.0.0)
Requirement already satisfied: py4j in ./.local/lib/python3.8/site-packages (0.10.9.7)
Requirement already satisfied: pyspark in ./.local/lib/python3.8/site-packages (3.3.1)
```

Figure 3 Installing python3, Pyspark and Jupyter

6. We will now install Jupyter Notebook in addition to Python 3 to organise all data projects in one location and make it simpler to demonstrate the process.

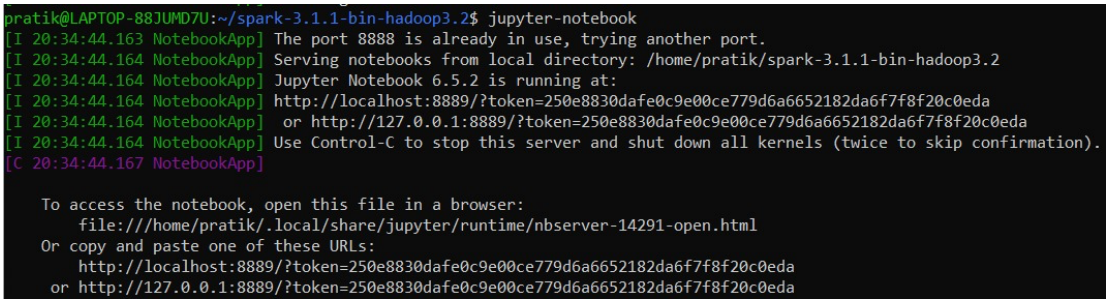
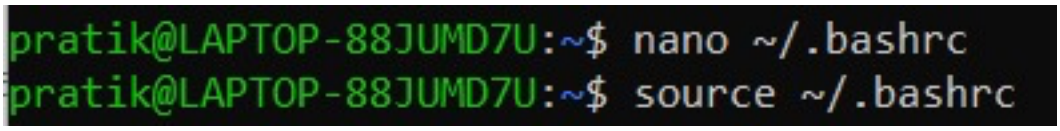


Figure 4 Jupyter Notebook in use

7. Additionally, we wll focus on "bashrc," a shell script that only starts interactive shell sessions. A file called "nano /.bashrc" contains settings for the Bash shell. For configuration files, the nano text editor is simple to use. The nano enables us to add, remove, or modify./bashrc file settings.Furthermore, we wll focus on "bashrc," a shell script that only starts interactive shell sessions. A file called "nano /.bashrc" contains settings for the Bash shell. For configuration files, the nano text editor is simple to use. The nano enables us to add, remove, or modify./bashrc file settings.



8. We will set the alias for Jupyter Notebook in this stage. It may be simpler to recall the command to start Jupyter Notebook if it has an alias. We must create an alias for the Jupyter notebook in a nano file after installing Python 3 and Java.

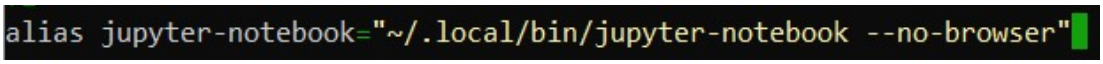


Figure 5 assigning alias to the jupyter notebook

Scala, Spark, and Hadoop will be installed in the last stage of the installation process. Ubuntu will immediately install the most recent version of Scala by just running the command from its terminal. The reason why downloading it is the simplest option is that the terminal will install and extract the file without the requirement for folder extraction.
The most important part now is to use "nano /.bashrc" to set Scala's path.

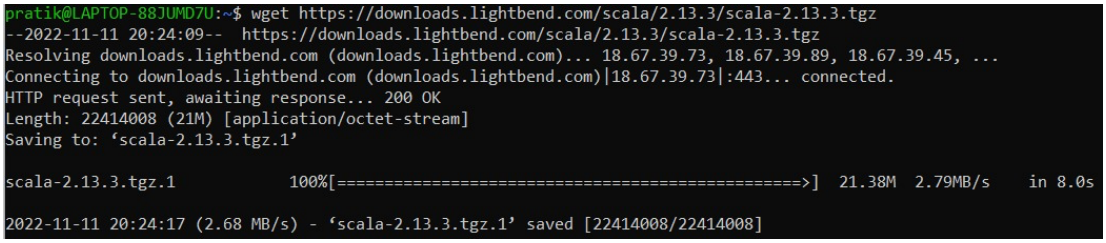
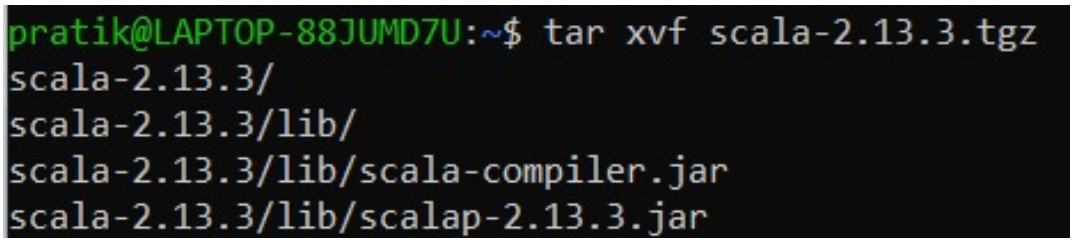


Figure 6 Downloading Scala

The export command should be used to set environment variables. This makes it very likely that the Spark programme can be identified and used.



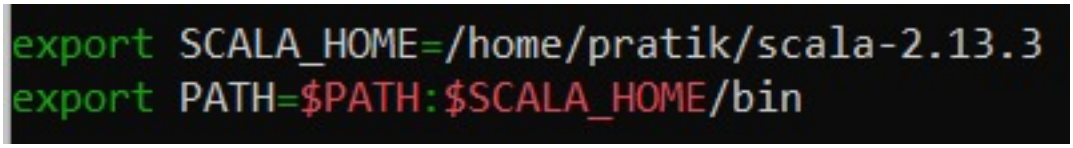


Figure 7 path setup in neno file

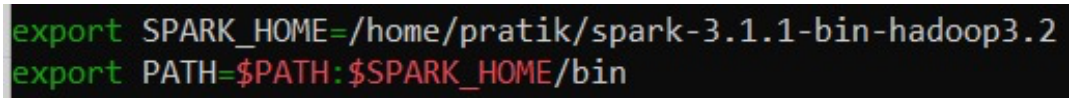


Figure 8 path setup for spark and hadoop

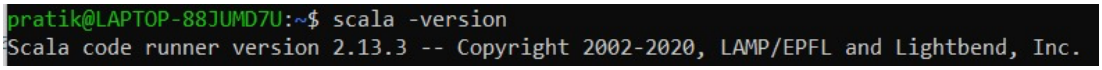


Figure 9 checking the scala version

The source /.bashrc command updates the current shell's configurations using the settings from the.bashrc file. This comes in handy if you want to alter something in the ~/.bashrc file and want it to take effect right away without having to close and reopen your terminal.

9. The command line interface for working with Spark is called Spark-shell. It offers a straightforward method of accessing the Spark UI and submitting Spark jobs. To run Spark commands and applications interactively, use the Spark shell. It has an integrated Scala interpreter and enables interactive data querying.

```
pratik@LAPTOP-88JUMD7U:~/spark-3.1.1-bin-hadoop3.2$  spark-shell
22/11/11 20:32:36 WARN Utils: Your hostname, LAPTOP-88JUMD7U resolves to a loopback address: 127.0.1.1; using 172.31.191
.13 instead (on interface eth0)
22/11/11 20:32:36 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/pratik/spark-3.1.1-bin-hadoop3.2/jars
/spark-unsafe_2.12-3.1.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
22/11/11 20:32:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://172.31.191.13:4040
Spark context available as 'sc' (master = local[*], app id = local-1668216763496).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 11.0.17)
Type in expressions to have them evaluated.
Type :help for more information.
```

Figure 10 Spark shell

10. We will be given some links to the notebook after running code on the Jupyter notebook. We can now access Jupyter Lab by using these URLs, which implies that whenever we need to run a command, the Ubuntu terminal will open the Jupyter notebook.

```
pratik@LAPTOP-88JUMD7U:~$ wget https://archive.apache.org/dist/spark/spark-3.1.1/spark-3.1.1-bin-hadoop3.2.tgz
--2022-11-11 20:28:32--  https://archive.apache.org/dist/spark/spark-3.1.1/spark-3.1.1-bin-hadoop3.2.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 228721937 (218M) [application/x-gzip]
Saving to: ‘spark-3.1.1-bin-hadoop3.2.tgz.1’
```

Figure 11 Jupyter notebook in spark

```
In [2]: from pyspark import SparkContext

In [3]: sc = SparkContext()

22/11/11 20:37:38 WARN Utils: Your hostname, LAPTOP-88JUMD7U resolves to a loopback address: 127.0.1.1; using 172.31.191.13 ins
tead (on interface eth0)
22/11/11 20:37:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/pratik/spark-3.1.1-bin-hadoop3.2/jars/spark-
unsafe_2.12-3.1.1.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
22/11/11 20:37:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes w
here applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

In [4]: sc

Out[4]: SparkContext

Spark UI
Version
v3.1.1
Master
local[*]
AppName
pyspark-shell
```

Figure 12 Jupyter notebook and version

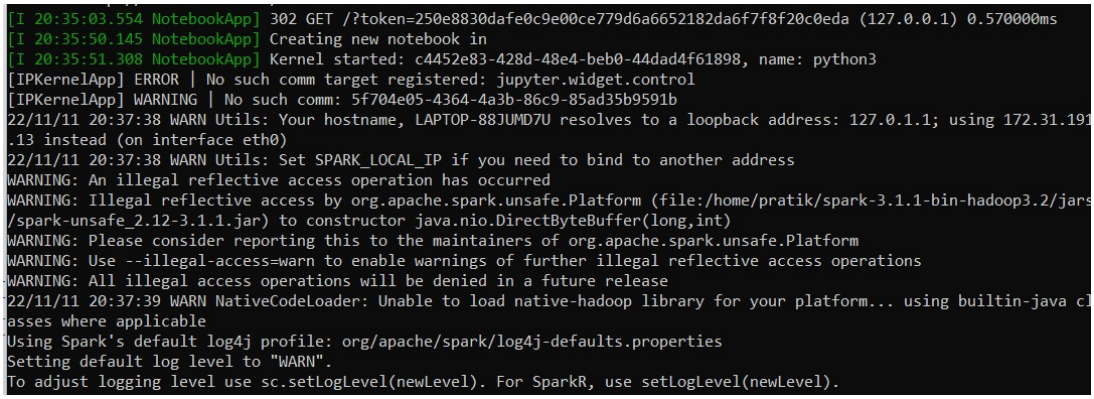


Figure 13 Logs

11. Finally, it becomes to stop the master.sh and worker.sh

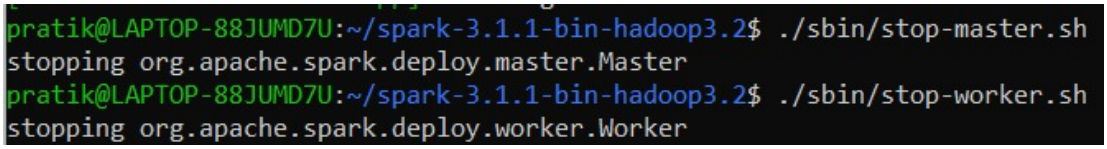


Figure 14 Stapping the master and worker.sh

References

Windows 10 tutorial: install WSL2 — Windows Subsystem for Linux 2. (2020, November 11). YouTube. Retrieved November 10, 2022, from https://www.youtube.com/watch?v=n-J9438Mv-s