Text

Description automatically generated

Toronto, Canada

**A Report on**

Lab 0: Grab Azure VM or Install VMWare and Start Cloudera

Data Management and Big Data

(ALY 6110)

Guided by:

Prof. Mohammad Shafiqul Islam

|  |  |  |
| --- | --- | --- |
| Name of Student | Submitted By:  NUID | Date of submission |
| Parth Shah | 002956963 | 11th November, 2022 |

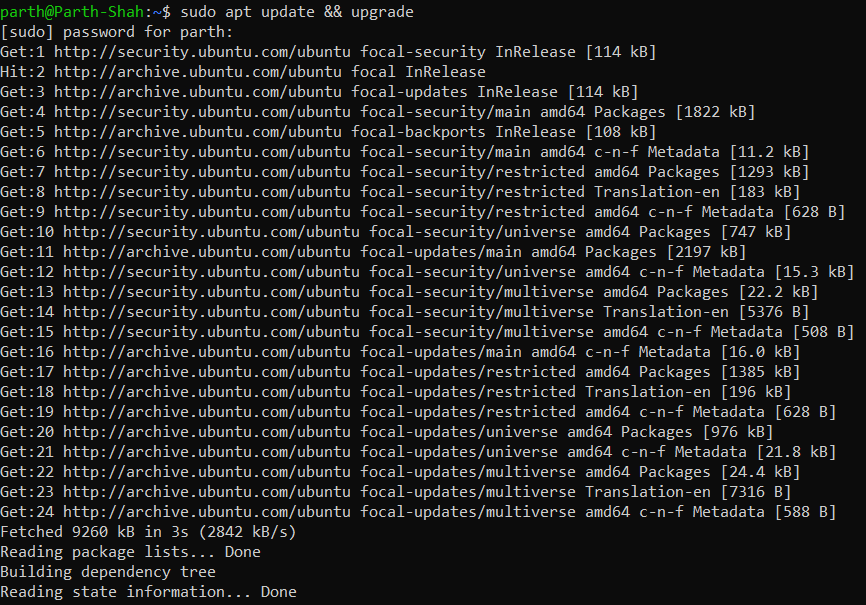
**Introduction**

Using Ubuntu Linux and its commands, we are performing Single Node Cluster in this Lab 0. To run Ubuntu Linux, WSL2.0 is installed as a virtual machine. To perform all the queries for this lab, we utilised the Ubuntu terminal. The commands sudo, apt, update, upgrade, tar, wget, install, pip3, nano, source,.bashrc, and version were among those we used. Here, we're using PySpark and a Jupyter notebook to build a single node cluster. The report's subsequent sections will cover the specific steps and commands.

**Steps to create Spark Cluster**

To create a PySpark cluster, we must first install the Java Run-Time Environment (JRE) and Java Development Kit (JDK). Here are the steps for installation:

**Updating and upgrading the system**



**Figure 1: Update and upgrade**

Sudo here refers to superuser do. Only superusers are authorised to perform the sudo command. APT, which stands for advanced package tool, is a group of tools for managing, inserting, and updating other software packages. The upgrade command is used to obtain the most recent versions of all the packages, whereas the update command is used to resynchronize the packages.

**Installing JRE and JDK**

Text

Description automatically generated

Figure 2: JRE and JDK installed

For the cluster creation procedure, JDK, the Java Development Kit, and JRE, the Java Run-Time Environment, are installed here.

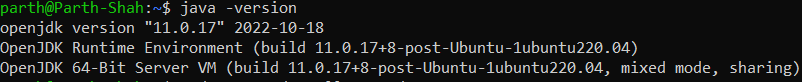


Figure 3: Checking Java version

Here, we are checking java version through “java -version” command. Output image shows that JDK version is 11.0.17 and JRE is built in the system terminal.

**Installing Python**

Text

Description automatically generated

Figure 4: Updating and upgrading all packages

Text

Description automatically generated

Figure 5: Installing pip3 and python 3

Here, we first update and upgrade every package, as seen in image number 4. Second, to launch PySpark and the Jupyter notebook, we installed pip3 and Python 3. Python 3 uses the pip3 package management and pip command. Python package index (PypI) installation is performed by Pip3.

Background pattern

Description automatically generated

Figure 6: Installing Python interpreter

Here, the py4j Python interpreter is being installed. To access Java objects in the Java Virtual Machine, Py4j is needed. This gives us access to Java virtual machine objects.



Figure 7: Setting alias

In this case, we are using the nano editor to set an alias for a Jupyter notebook. A script file called ".bashrc" is executed when a user logs into the system. The phrase "Source /. Bashrc" denotes that the already-edited shell environment will be preserved.

****

Figure 8: Setting path for jupyter notebook

We are building up the pathway to the nano editor for the Jupyter notebook.

**Installing Scala**

Text

Description automatically generated

Figure 9: Installing and Extracting Scala

Here, we are installing Scala. To install Scala, Wget command is used. Through Wget command and link at suffix can download the required package. Through Tar command, we can extract the downloaded package. Here, “xvf” is extract, verbose, file.



Figure 10: setting alias for Scala

In this section, we configure Scala's path into the nano editor. Our changes are being made to the .bashrc file.



Figure 11: Identifying Scala version

Here, by running “scala -version”, we can identify the version. Version of Scala is 2.11.12.

**Installing Spark with Hadoop**

Text

Description automatically generated

Figure 12: Installing Spark with Hadoop

Here, Spark with Hadoop are being installed. Wget is used to install Spark with Hadoop. The necessary package may be downloaded using the Wget command and the URL at the suffix. We can extract the downloaded package using the Tar programme. Here, "xvf" stands for extract, verbose, file.



Figure 13: Setting up path For Spark with Hadoop

We configure Spark's route into the nano editor in this part. We're making modifications to the.bashrc file.

**Activating Spark**

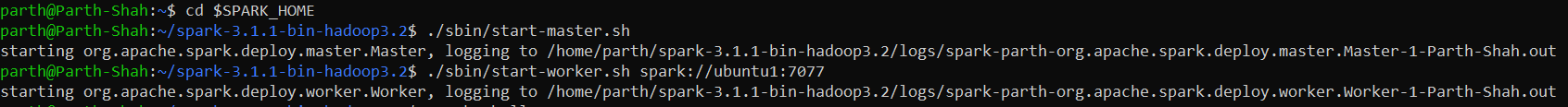


Figure 14: Starting Master and Worker spark

For activating Spark shell, we already set the path in the .bashrc file. Firstly, we will go into the SPARK\_HOME and we will start master and worker shell to activate the Spark shell.

Text

Description automatically generated

Figure 15: Spark Shell

The Spark shell is seen in the terminal in the image above. Spark is active and shows Spark authored and the version when the Spark shell is entered into the command line interface. The Spark is at version 3.1.1.

**Testing PySpark using Jupyter Notebook**

Text

Description automatically generated

Figure 16: Accessing Jupyter Notebook

When we type the command "jupyter-notebook," we will receive the URL that must be copied and pasted into the browser to access the Jupyter Notebook.

Graphical user interface, text, application, Word

Description automatically generated

Figure 17: Testing PySpark

We are importing Spark Context from PySpark in this instance. We are giving the SparkContext() the SC name. We can obtain the spark version, Master branch, and AppName, which is pyspark-shell, by executing "SC".

**Stopping Cluster**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 18: Stopping cluster

Here, we are stopping the cluster of master and worker branch in the terminal.

**Reference**

Islam, S. (2022, October 23). *Single Node Spark/PySpark Cluster on Windows Subsystem for Linux (WSL2)*. Medium. Retrieved November 11, 2022, from <https://medium.com/mlearning-ai/single-node-spark-pyspark-cluster-on-windows-subsystem-for-linux-wsl2-22860888a98d>