APACHE SPARK

1. Download Java

Java –version (to see if java is installed in the system)

sudo apt-get install default-jdk (to download java)

udo update-alternatives --config java(to change java version)

1. Download Spark Package

Youtube - <https://youtu.be/YanzUI-30pI>

spark website - <https://spark.apache.org/downloads.html>

hadoop 3.3 and later (Scala 2.13)

go to download page and copy the link

go to terminal

(go to root user) sudo su –

Create a directory in root to download spark)

mkdir –p /opt/spark

cd opt/spark

wget <paste the link> <https://dlcdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2-scala2.13.tgz>

(untar the tgz file that you get)tar –xvz <tar file>

/opt/spark/ spark-3.2.1-bin-hadoop3.2-scala2.13

1. Setting Environment variable

Sudo nano ~/.bashrc

In the end of the file insert-

export SPARK\_HOME=/opt/spark/<insert the untar file over here>

export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin

Activate the Changes

source ~/.bashrc

1. Verification of Spark Installation

Spark-shell

Scala> val num=Array(1,2,3,56,78)

Scala>val data = sc.parallelize(num)

Scala>data.foreach(println)

Creating SparkSessions

object SparkSessionTest extends App{val Spark=SparkSession.builder().appName("SparkByExamples").getOrCreate()}

creating dataset using rdd

val rdd=spark.sparkContext.parallelize(Seq((1,"Java"),(2,"Python")))

val dataset=rdd.toDS()

merging datasets

val resultDS=dataset.union(dataset2)

***how to create dataframes and store them as csv files***

val data=Seq(("Sakhib","Rahil","USA","CA"),("Staline","DSouza","U

SA","NY"), ("Guru","Kajgar","USA","CA"),("Maria","Naidu","USA","FL") )

val columns = Seq("firstname","lastname","country","state")

import spark.implicits.\_

val df = data.toDF(columns:\_\*)

df.write.csv("/tmp/spark\_output/csv")

/home/azureuser/sakhib

How to read data from json files?

**val** DF **=** spark.read.format("json").option("sep", ";").option("inferSchema", "true").option("header", "true") .load(<path>)

Create sqlContext

val sqlContext = new org.apache.spark.sql.SQLContext(sc)

val dfs=sqlContext.read.json("/tmp/spark\_output/json")

/root/tmp/spark\_output/Jai

JAVA\_HOME: /usr/lib/jvm/java-11-openjdk-amd64

To start spark with all jar files

spark-shell --jars /opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar

spark-shell --jars /opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar, /opt/spark/azure-storage-blob-12.10.2.jar

/opt/spark/jetty-util-9.3.24.v20180605.jar

spark.jars.com

**val** conf **=** **new** **SparkConf**().set("spark.hadoop.abc.def", "xyz")

**val** sc **=** **new** **SparkContext**(conf)

val spark = SparkSession.builder()

.master("local[1]")

.appName("SparkByExamples.com")

.getOrCreate();

val conf = new SparkConf().set("spark.jars","/opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar")

val sc = new SparkContext(conf)

spark.sparkContext.listJars.foreach(println)

org.apache.spark.sql.SparkSession.builder().master("local[1]").appName("SparkByExamples.com").config("fs.azure.account.key.training005.blob.core.windows.net", "5lM/dAr/sdPbLepuszg2kyJvu21sqb5PbVyrsmf6P8YRjgJuBUsdSv4z0C2ZSRflWrkeGJ/xQSBp+AStRA+RxQ==").config("spark.jars", "/opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar").getOrCreate()

df.write.json(s"wasbs://training005-container@training005.blob.core.windows.net/Sakhib")

hadoop-azure

<https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-azure/3.3.3/hadoop-azure-3.3.3.jar>

azure-storage

<https://repo1.maven.org/maven2/com/microsoft/azure/azure-storage/8.6.6/azure-storage-8.6.6.jar>

Pom.xml file

<!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-azure -->

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-azure</artifactId>

<version>3.3.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.microsoft.azure/azure-storage -->

<dependency>

<groupId>com.microsoft.azure</groupId>

<artifactId>azure-storage</artifactId>

<version>8.6.6</version>

</dependency>

<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<configuration><property>  
 <name>fs.AbstractFileSystem.wasb.Impl</name>  
 <value>org.apache.hadoop.fs.azure.Wasb</value>  
 </property> <property>  
 <name>fs.azure.account.key.**StorageAccountName**.blob.core.windows.net</name>  
 <value>xxxxx....</value> //**AccessKey**   
 </property> <property>  
 <name>fs.azure.block.blob.with.compaction.dir</name>  
 <value>/hbase/WALs,/data/myblobfiles</value>  
 </property> <property>  
 <name>fs.azure</name> <value>org.apache.hadoop.fs.azure.NativeAzureFileSystem</value>  
 </property>  
   
<property>  
 <name>fs.azure.enable.append.support</name>  
 <value>true</value>  
 </property></configuration>

spark-shell --jars /opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar, /opt/spark/azure-storage-blob-12.10.2.jar, /opt/spark/azure-keyvault-core-1.2.6.jar,

/opt/spark/jetty/jetty-util-9.3.24.v20180605.jar, /opt/spark/jetty/jetty-http-11.0.11.jar,/opt/spark/jetty/jetty-io-11.0.11.jar, /opt/spark/jetty/jetty-security-11.0.11.jar,/opt/spark/jetty/jetty-servlet-11.0.11.jar, /opt/spark/jetty/jetty-webapp-11.0.11.jar,/opt/spark/jetty/jetty-server-11.0.11.jar, /opt/spark/jetty/jetty-xml-11.0.11.jar

FINAL SPARK SHELL

spark-shell --jars /opt/spark/azure-storage-8.6.4.jar,/opt/spark/hadoop-azure-3.2.1.jar,/opt/spark/azure-storage-blob-12.10.2.jar,/opt/spark/azure-keyvault-core-1.2.6.jar,/opt/spark/jetty/jetty-util-9.3.24.v20180605.jar,/opt/spark/jetty/jetty-http-11.0.11.jar,/opt/spark/jetty/jetty-io-11.0.11.jar,/opt/spark/jetty/jetty-security-11.0.11.jar,/opt/spark/jetty/jetty-servlet-11.0.11.jar,/opt/spark/jetty/jetty-webapp-11.0.11.jar,/opt/spark/jetty/jetty-server-11.0.11.jar,/opt/spark/jetty/jetty-xml-11.0.11.jar

**package** com.youTube  
**import** org.apache.spark.\_  
  
**object** WordCount  
{  
 **def** main(args:Array[String]): Unit =  
 {  
 **val** conf = **new** SparkConf()  
 conf.set(**"spark.master"**,args(0))  
 conf.set(**"spark.app.name"**,args(1))  
 **val** sc = **new** SparkContext(conf)  
 sc.setLogLevel(**"WARN"**)  
 **val** r1 = sc.textFile(args(2))  
 **val** r2 = r1.flatMap(\_.split(**' '**)).map((\_,1)).reduceByKey(\_\*\_)  
 r2.saveAsTextFile(args(3))  
 sc.stop()  
 }  
}

HOW TO TRANSFER DATA FROM LOCAL MACHINE TO AZURE LINUX VM?

scp <file location> ssh [azureuser@40.80.82.167:<destination\_location](mailto:azureuser@40.80.82.167:%3cdestination_location)>

export JAVA\_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java"

HOW TO CHANGE JAVA VERSION

sudo update-alternatives --config java

/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

Mode:overwrite

.config(**"fs.azure"**,**"org.apache.hadoop.fs.azure.NativeAzureFileSystem"**)  
.config(**"fs.azure.account.key.training005.blob.core.windows.net"**,  
 **"5lM/dAr/sdPbLepuszg2kyJvu21sqb5PbVyrsmf6P8YRjgJuBUsdSv4z0C2ZSRflWrkeGJ/xQSBp+AStRA+RxQ=="**)