Course: Big Data

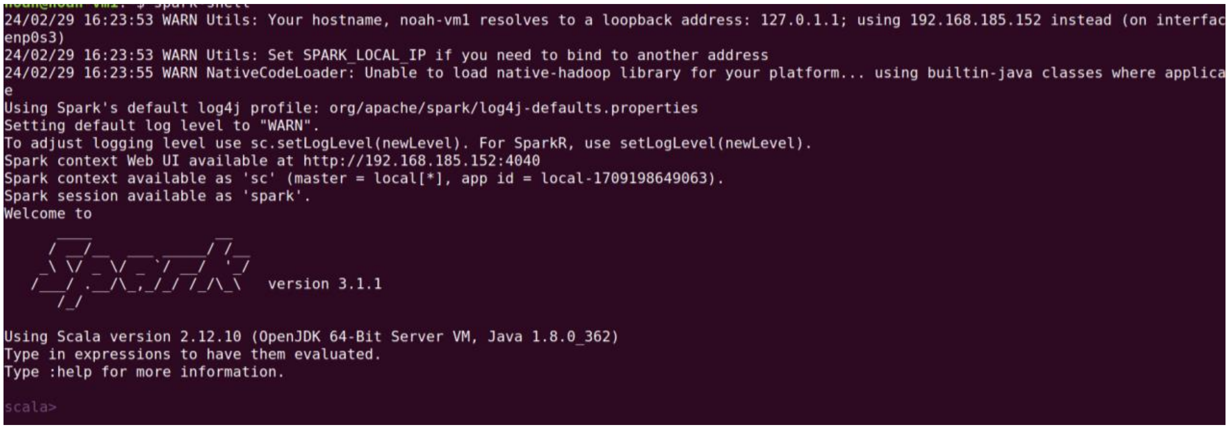
Lab 04

PySpark - RDD

Question 1:

Based on the tutorial of PySpark, students install PySpark in Ubuntu.

- Define the environment variable: JAVA_HOME
- Define the environment variable: SPARK_HOME
- Start the pyspark-shell and write an instruction to print down the PySpark version
- Take the screenshot and insert it into the table below.



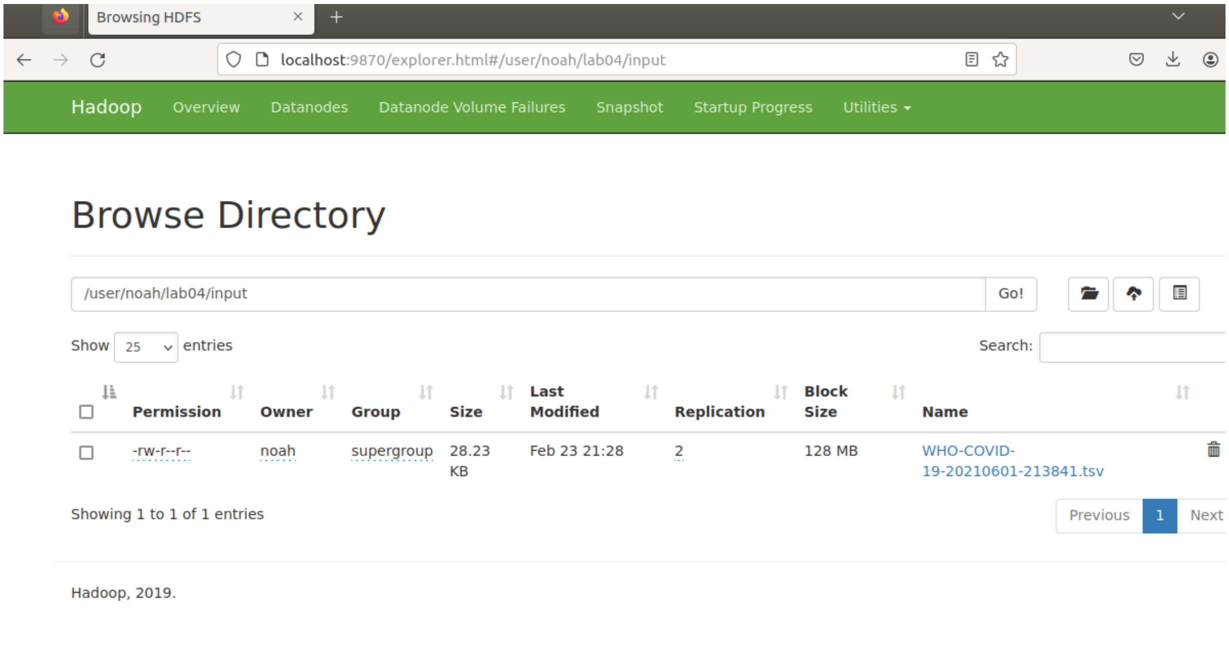
```
noah@noah-vm1:~$ spark-shell
24/02/29 16:23:53 WARN Utils: Your hostname, noah-vm1 resolves to a loopback address: 127.0.1.1; using 192.168.185.152 instead (on interfac
enp0s3)
24/02/29 16:23:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
24/02/29 16:23:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applica
e
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://192.168.185.152:4040
Spark context available as 'sc' (master = local[*], app id = local-1709198649063).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_362)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Question 2:

Given a tsv file WHO-COVID-19-20210601-213841.tsv which is corresponding to the WHO Coronavirus (COVID-19) Dashboard.

Students are required to create a folder, named **lab04**, in HDFS and then copy the tsv to **lab04/input/**

Take a screenshot to show the content of **lab04/input/** in HDFS



Browse Directory

/user/noah/lab04/input

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	noah	supergroup	28.23 KB	Feb 23 21:28	2	128 MB	WHO-COVID-19-20210601-213841.tsv

Showing 1 to 1 of 1 entries

Hadoop, 2019.

Question 3:

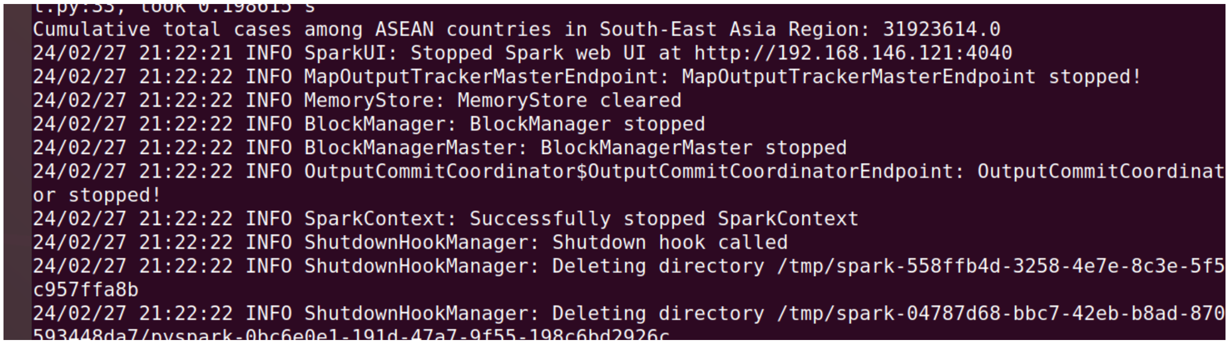
Write a PySpark program, located in **ASEANCaseCount.py**, to count the number of cumulative total cases among ASEAN countries (*South-East Asia Region in the given data table*) using RDDs.

- Insert your source code into the table below.

```
from pyspark.sql import SparkSession

def filter_south_east_asia(line):
parts = line.split('\t')
region = parts[1].strip()
return region == "South-East Asia"

def parse_and_filter(line):
parts = line.split('\t')
country = parts[0]
cumulative_total_str = parts[2].replace(',','')
cumulative_total = float(cumulative_total_str)
return country, cumulative_total

if __name__ == "__main__":
# Create instance of SparkSession
spark = SparkSession.builder.appName("ASEANCaseCount").getOrCreate()

input_path = 'hdfs://localhost:9000/user/hoang/lab04/input/WHO-COVID-19-20210601-
```

```
213841.tsv'

input_rdd = spark.sparkContext.textFile(input_path)

filtered_rdd = input_rdd.filter(filter_south_east_asia)

asean_rdd = filtered_rdd.map(parse_and_filter)
print(asean_rdd.count())

cumulative_total = asean_rdd.map(lambda x: x[1]).reduce(lambda x, y: x + y)

print("Cumulative total cases among ASEAN countries in South-East Asia Region:",
cumulative_total)

spark.stop()
```

- Take a screenshot of the terminal to visualize the program result.
Result: Cumulative total cases among ASEAN countries in South-East Asia Region: 31923614 cases.



```
t.py:33, took 0.198613 s
Cumulative total cases among ASEAN countries in South-East Asia Region: 31923614.0
24/02/27 21:22:21 INFO SparkUI: Stopped Spark web UI at http://192.168.146.121:4040
24/02/27 21:22:22 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/02/27 21:22:22 INFO MemoryStore: MemoryStore cleared
24/02/27 21:22:22 INFO BlockManager: BlockManager stopped
24/02/27 21:22:22 INFO BlockManagerMaster: BlockManagerMaster stopped
24/02/27 21:22:22 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinat
or stopped!
24/02/27 21:22:22 INFO SparkContext: Successfully stopped SparkContext
24/02/27 21:22:22 INFO ShutdownHookManager: Shutdown hook called
24/02/27 21:22:22 INFO ShutdownHookManager: Deleting directory /tmp/spark-558ffb4d-3258-4e7e-8c3e-5f5
c957ffa8b
24/02/27 21:22:22 INFO ShutdownHookManager: Deleting directory /tmp/spark-04787d68-bbc7-42eb-b8ad-870
593448da7/pyspark-0bc6e0e1-191d-47a7-9f55-198c6bd2926c
```