RT4 · Big Data Class Labs

Lab 1 · Batch processing with Hadoop HDFS and Map Reduce

Adam Lahbib · M. Sofiene Barka · Mohamed Rafraf

Mar. 1, 2023 @ INSAT

Contents

1 Introduction 4

2 Definitions 5

2.1 Hadoop 5

2.2 HDFS 5

2.3 MapReduce 5

2.4 Java MapReduce API 5

2.5 Docker 6

3 Environment 7

3.1 WSL2 (Ubuntu) + Docker CLI + Hadoop image (Liliasfaxi's Image in DockerHub) 7

3.2 JDK 8 + JDK 11 8

3.3 Installing Maven 9

3.4 Docker Network 10

3.5 Master and 2 slaves 11

3.6 Starting Hadoop 11

3.7 First Steps with Hadoop 12

3.8 Hadoop Web UI 12

4 Application 15

4.1 Idea 15

4.2 Code Modifications 15

4.2.1 `WordCount.java` 15

4.2.2 `TokenizerMapper.java` 16

4.2.3 `IntSumReducer.java` 16

4.2.4 Github repo for source code: https://github.com/adamlahbib/BigData_Labs/tree/main/lab1/wordcount 17

4.2.5 `pom.xml` File 17

4.3 Local test using VSCode 18

4.3.1 Packages Installation: `mvn package install` 20

4.4 Hadoop test (Dockerized) 20

4.5 Monitoring the job 24

5 Homework 27

5.1 Idea 27

5.2 Architecture 27

5.3 Code 27

5.3.1 `WordCount.java` 27

5.3.2 `TokenizerMapper.java` 28

5.3.3 `IntSumReducer.java` 30

5.4 Results 31

6 Conclusion 33

1 Introduction

In this lab, we will learn how to use Hadoop to process large amounts of data. We will use the Hadoop Distributed File System (HDFS) to store the data and the MapReduce framework to process it. We will use the Java MapReduce API to write our MapReduce programs.

2 Definitions

2.1 Hadoop

Hadoop is an open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled in software by the framework.

2.2 HDFS

HDFS is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

2.3 MapReduce

MapReduce is a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster. The MapReduce programming model is comprised of the map and reduce operations. The map operation processes a set of input key/value pairs to generate a set of intermediate key/value pairs. The reduce operation merges all intermediate values associated with the same intermediate key to produce a set of output key/value pairs.

2.4 Java MapReduce API

The Java MapReduce API is a Java programming interface for writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

2.5 Docker

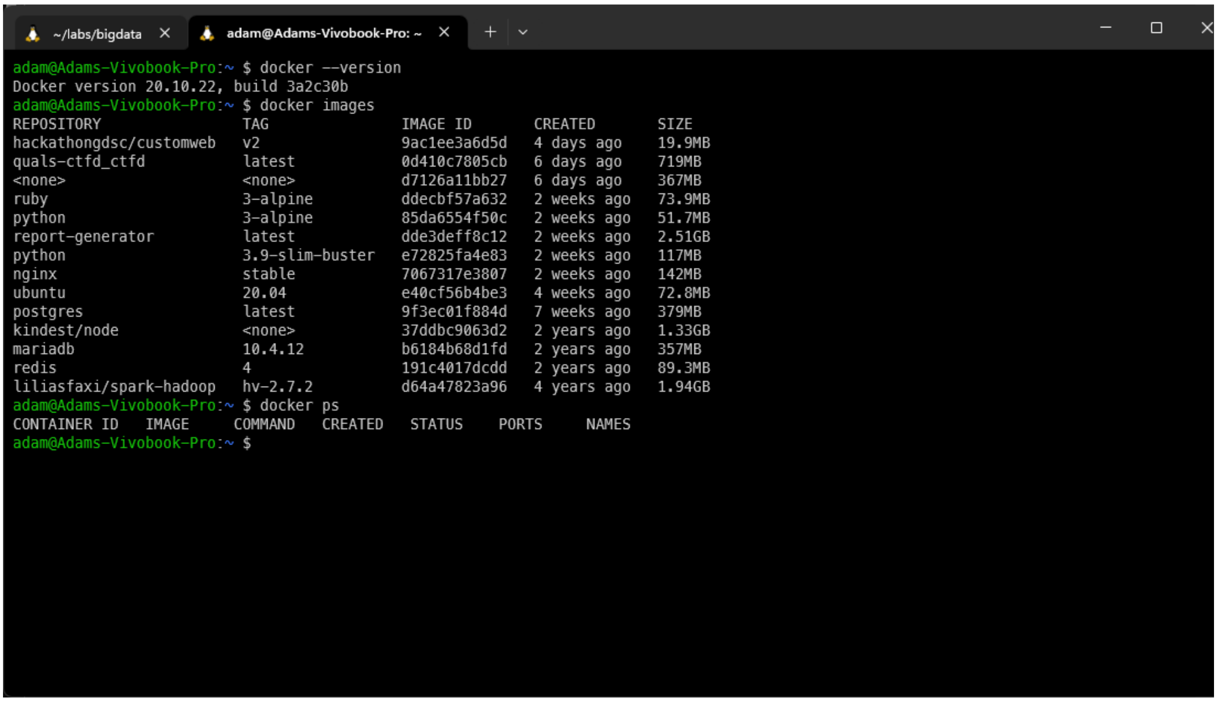
Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating-system kernel and are thus more lightweight than virtual machines. Containers are created from images that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories.

3 Environment

Requirements:

- Docker (latest)
- JDK 8
- Maven 4.0.0
- WSL2 (Ubuntu/Debian preferrably) or Linux Distro
- Hadoop image (latest)

3.1 WSL2 (Ubuntu) + Docker CLI + Hadoop image (Liliasfaxi's Image in DockerHub)



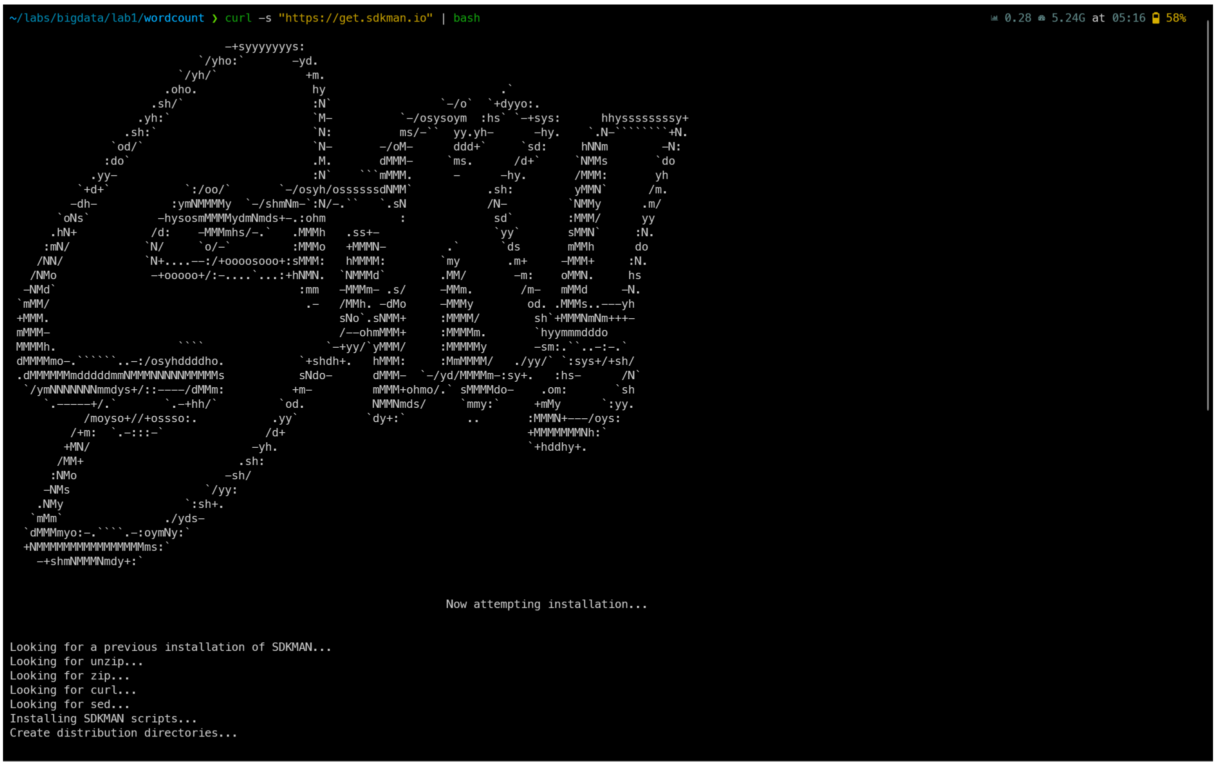
```
adam@Adams-Vivobook-Pro:~ $ docker --version
Docker version 20.10.22, build 3a2c30b
adam@Adams-Vivobook-Pro:~ $ docker images
REPOSITORY                 TAG               IMAGE ID       CREATED       SIZE
hackathongdsc/customweb    v2                9ac1ee3a6d5d   4 days ago    19.9MB
quals-ctfd_ctfd            latest            0d410c7805cb   6 days ago    719MB
<none>                     <none>            d7126a11bb27   6 days ago    367MB
ruby                       3-alpine          ddecbf57a632   2 weeks ago   73.9MB
python                     3-alpine          85da6554f50c   2 weeks ago   51.7MB
report-generator           latest            dde3deff8c12   2 weeks ago   2.51GB
python                     3.9-slim-buster   e72825fa4e83   2 weeks ago   117MB
nginx                      stable            7067317e3807   2 weeks ago   142MB
ubuntu                     20.04             e40cf56b4be3   4 weeks ago   72.8MB
postgres                   latest            9f3ec01f884d   7 weeks ago   379MB
kindest/node               <none>            37ddbc9063d2   2 years ago   1.33GB
mariadb                    10.4.12           b6184b68d1fd   2 years ago   357MB
redis                      4                 191c4017dcdd   2 years ago   89.3MB
liliasfaxi/spark-hadoop    hv-2.7.2          d64a47823a96   4 years ago   1.94GB
adam@Adams-Vivobook-Pro:~ $ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
adam@Adams-Vivobook-Pro:~ $
```

WSL 2 is a new version of the Windows Subsystem for Linux (WSL) that is based on a real Linux kernel, using a lightweight utility virtual machine (VM) manager called Hyper-V. WSL 2 is a significant improvement over WSL 1, which was based on a Linux compatibility layer called the Windows Subsystem for Linux (WSL) that was built into the Windows 10 kernel. WSL 2 is a significant improvement over WSL 1, which was based on a Linux compatibility layer called the Windows Subsystem for Linux (WSL) that was built into the Windows 10 kernel.

3.2 JDK 8 + JDK 11

JDK or Java Development Kit is a software development environment used for developing Java applications and applets. It contains the Java Runtime Environment, an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development. It is a development environment for building applications, applets, and components using the Java programming language. It includes an interpreter, a compiler, debuggers, and other tools needed to develop applications and applets that run in the Java runtime environment.

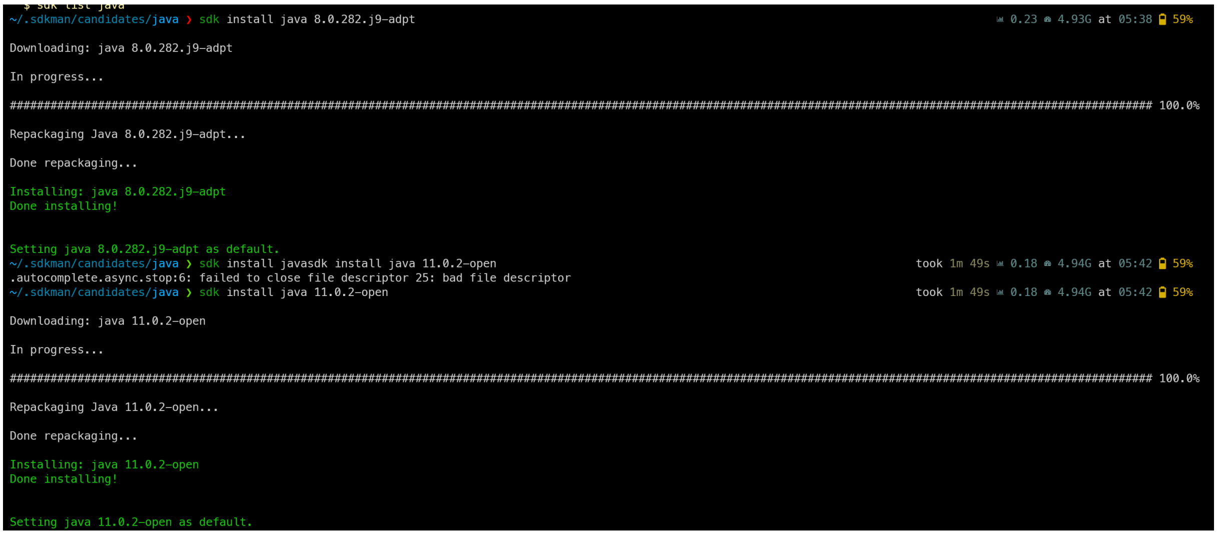
I installed SDKMan to manage my Java versions, it's a tool for managing parallel versions of multiple Software Development Kits on most Unix based systems. It provides a convenient Command Line Interface (CLI) and API for installing, switching, removing and listing Candidates. It is written in Groovy and uses Gradle for building. It is available under the Apache 2.0 license.



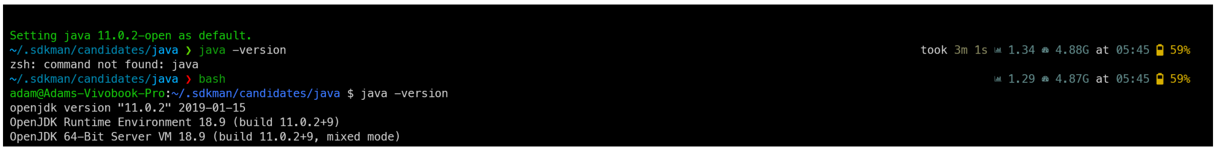
```
~/labs/bigdata/lab1/wordcount ❯ curl -s "https://get.sdkman.io" | bash                    0.28  5.24G at 05:16  58%

                                                    Now attempting installation...


Looking for a previous installation of SDKMAN...
Looking for unzip...
Looking for zip...
Looking for curl...
Looking for sed...
Installing SDKMAN scripts...
Create distribution directories...
```

`SDKMan list java`, can show all the available JDKs

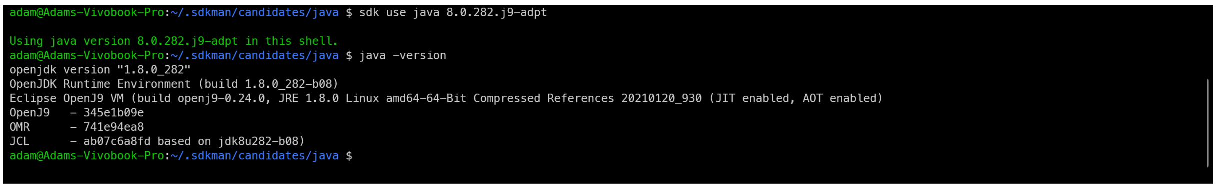
I installed Java JDK 8.0.302-open for this lab and Java JDK 11.0.2-open for myself, I'll keep JDK 11.0.2-open as my default Java version, and switch to JDK 8.0.302-open when I need to run Hadoop using `sdk use java xxx`



```
  $ sdk list java
~/.sdkman/candidates/java ❯ sdk install java 8.0.282.j9-adpt                                                                                0.23  4.93G at 05:38  59%

Downloading: java 8.0.282.j9-adpt

In progress...

######################################################################################################################################## 100.0%

Repackaging Java 8.0.282.j9-adpt...

Done repackaging...

Installing: java 8.0.282.j9-adpt
Done installing!


Setting java 8.0.282.j9-adpt as default.
~/.sdkman/candidates/java ❯ sdk install javasdk install java 11.0.2-open                                                          took 1m 49s  0.18  4.94G at 05:42  59%
.autocomplete.async.stop:6: failed to close file descriptor 25: bad file descriptor
~/.sdkman/candidates/java ❯ sdk install java 11.0.2-open                                                                          took 1m 49s  0.18  4.94G at 05:42  59%

Downloading: java 11.0.2-open

In progress...

######################################################################################################################################## 100.0%

Repackaging Java 11.0.2-open...

Done repackaging...

Installing: java 11.0.2-open
Done installing!


Setting java 11.0.2-open as default.
```



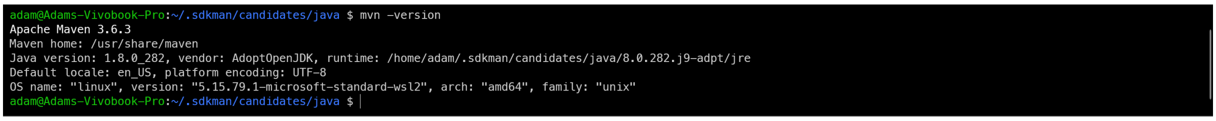
```


Setting java 11.0.2-open as default.
~/.sdkman/candidates/java ❯ java -version                                                                                          took 3m 1s  1.34  4.88G at 05:45  59%
zsh: command not found: java
~/.sdkman/candidates/java ❯ bash                                                                                                             1.29  4.87G at 05:45  59%
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $ java -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)
```



```
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $ sdk use java 8.0.282.j9-adpt

Using java version 8.0.282.j9-adpt in this shell.
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-b08)
Eclipse OpenJ9 VM (build openj9-0.24.0, JRE 1.8.0 Linux amd64-64-Bit Compressed References 20210120_930 (JIT enabled, AOT enabled)
OpenJ9   - 345e1b09e
OMR      - 741e94ea8
JCL      - ab07c6a8fd based on jdk8u282-b08)
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $
```

3.3 Installing Maven

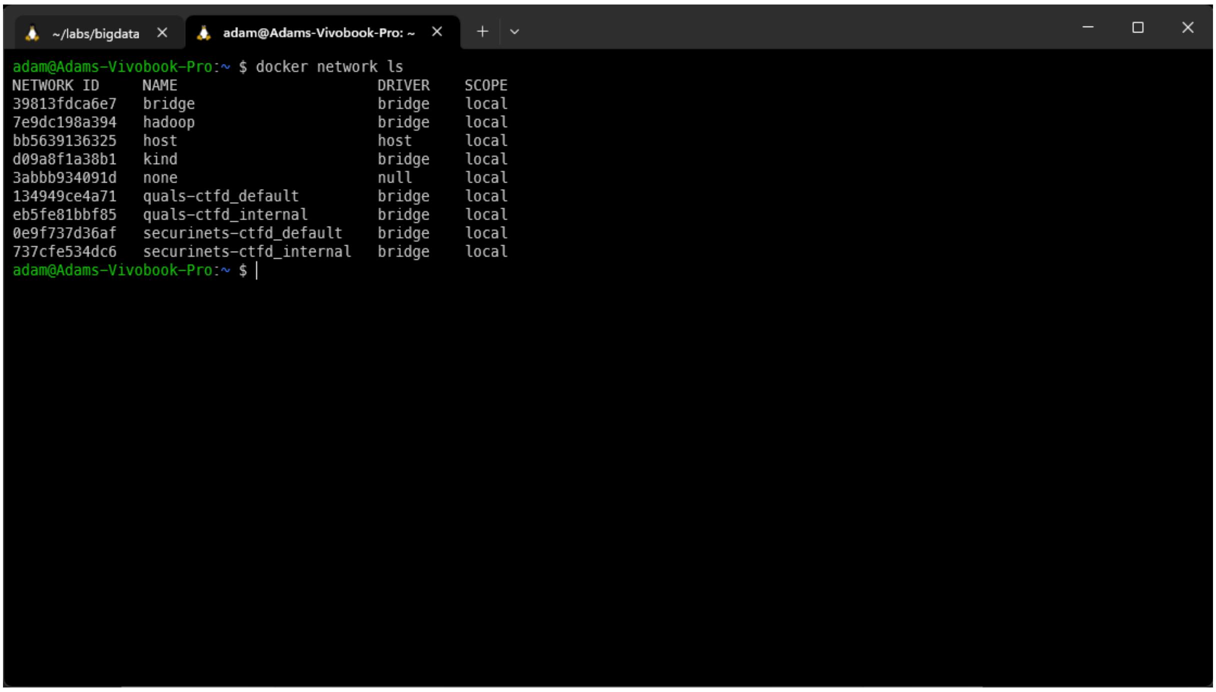
Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies. Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies.



```
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 1.8.0_282, vendor: AdoptOpenJDK, runtime: /home/adam/.sdkman/candidates/java/8.0.282.j9-adpt/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.15.79.1-microsoft-standard-wsl2", arch: "amd64", family: "unix"
adam@Adams-Vivobook-Pro:~/.sdkman/candidates/java $ |
```

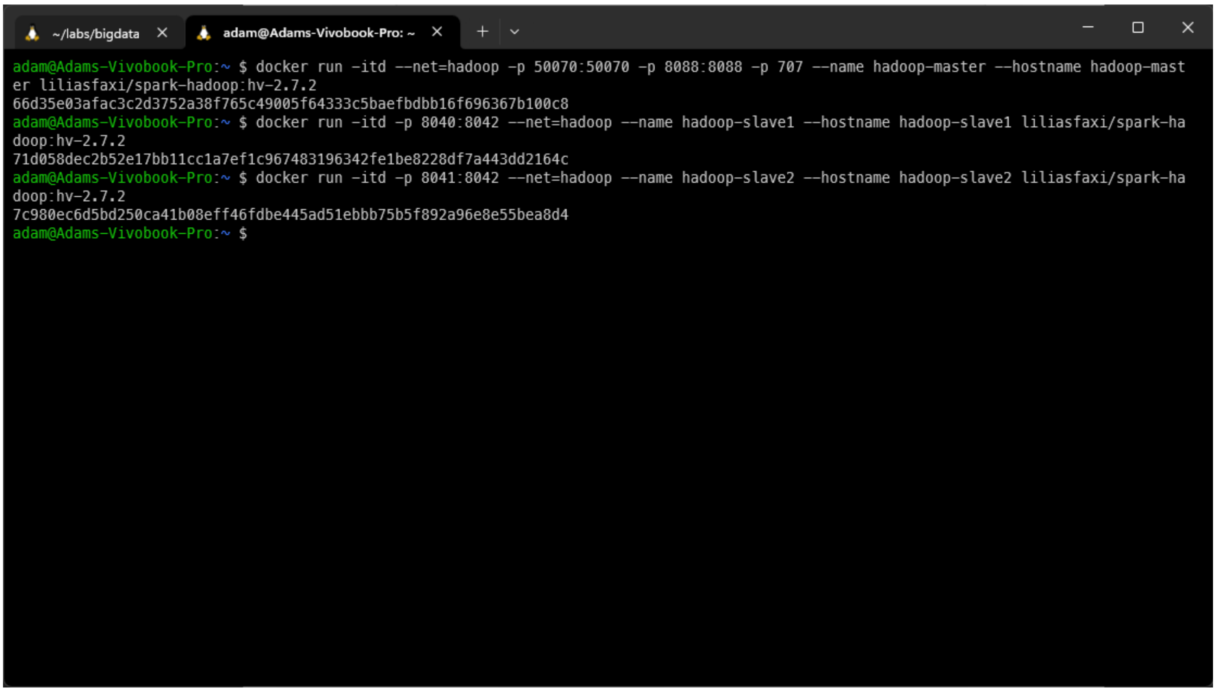
3.4 Docker Network

A Docker network is a virtual network that connects containers. It is a layer 2 bridge that connects containers running on the same Docker host. It is also a layer 3 network that connects containers running on different Docker hosts. Docker networks are isolated from each other and from the host by default. Docker networks are defined by a driver. The default driver is bridge. The bridge driver creates a virtual network that connects containers running on the same Docker host. The virtual network is isolated from the host network and other Docker networks by default. The bridge driver is the only driver that supports the default network. The bridge driver supports the creation of user-defined networks.



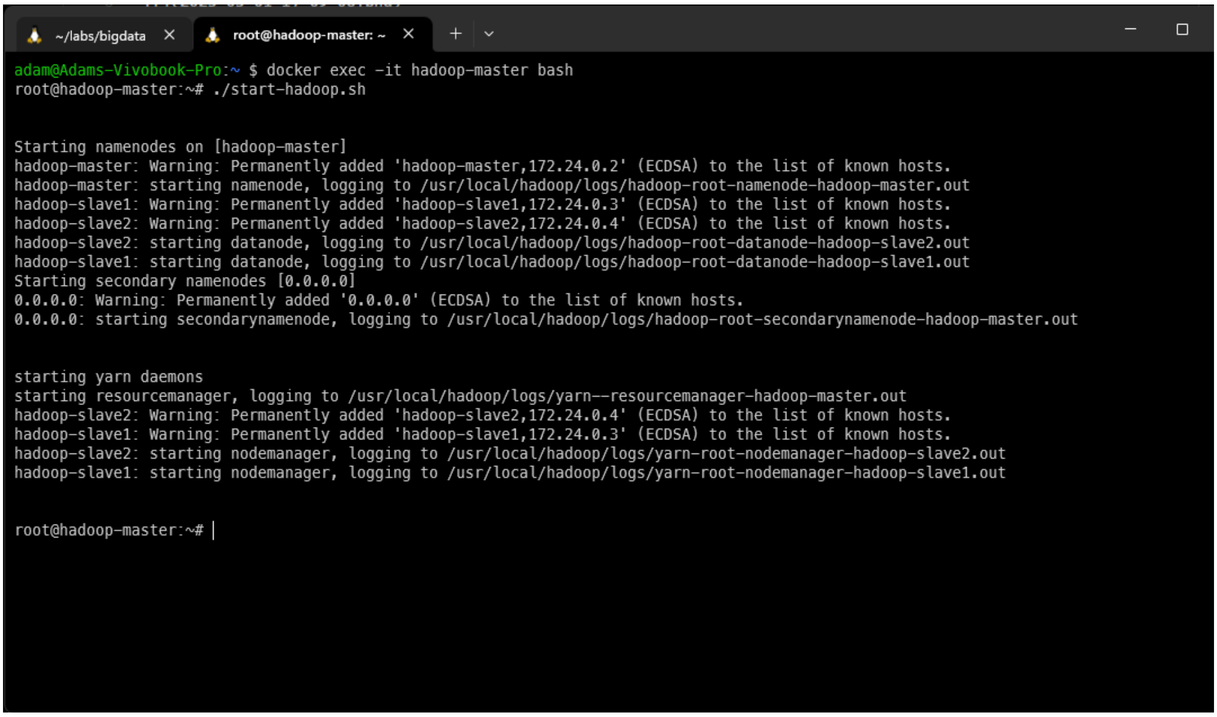
```
adam@Adams-Vivobook-Pro:~ $ docker network ls
NETWORK ID     NAME                        DRIVER    SCOPE
39813fdca6e7   bridge                      bridge    local
7e9dc198a394   hadoop                      bridge    local
bb5639136325   host                        host      local
d09a8f1a38b1   kind                        bridge    local
3abbb934091d   none                        null      local
134949ce4a71   quals-ctfd_default          bridge    local
eb5fe81bbf85   quals-ctfd_internal         bridge    local
0e9f737d36af   securinets-ctfd_default     bridge    local
737cfe534dc6   securinets-ctfd_internal    bridge    local
adam@Adams-Vivobook-Pro:~ $
```

3.5 Master and 2 slaves



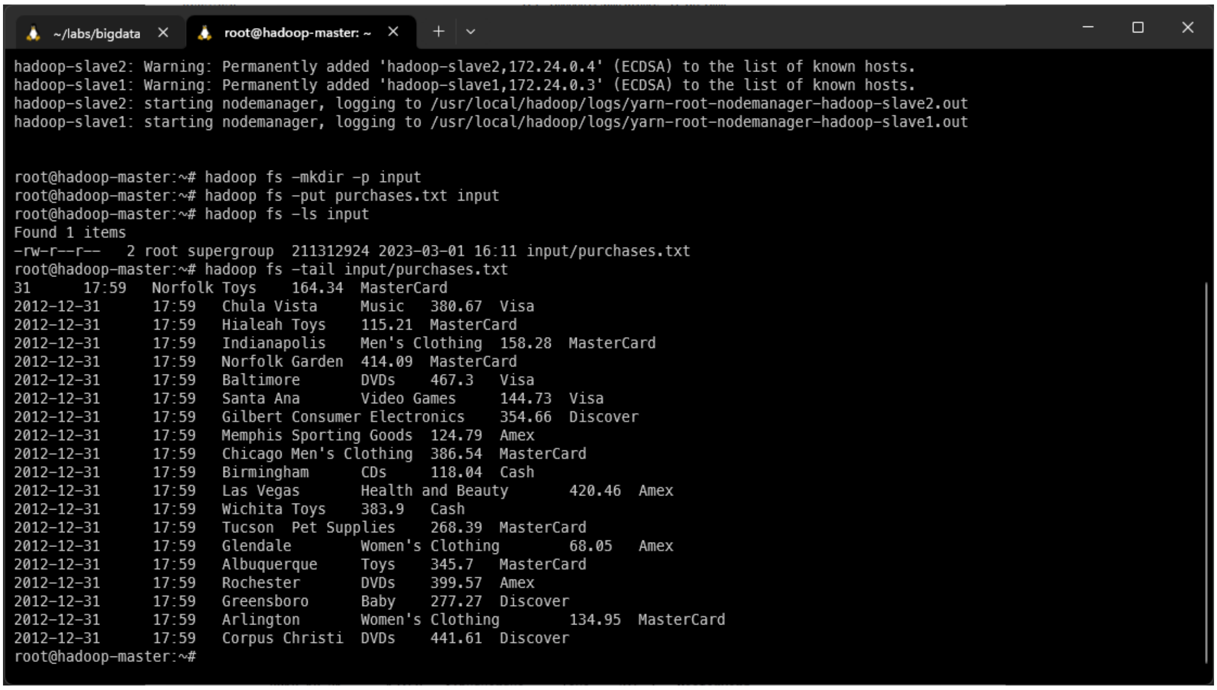
```
adam@Adams-Vivobook-Pro:~ $ docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 707 --name hadoop-master --hostname hadoop-mast
er liliasfaxi/spark-hadoop:hv-2.7.2
66d35e03afac3c2d3752a38f765c49005f64333c5baefbdbb16f696367b100c8
adam@Adams-Vivobook-Pro:~ $ docker run -itd -p 8040:8042 --net=hadoop --name hadoop-slave1 --hostname hadoop-slave1 liliasfaxi/spark-ha
doop:hv-2.7.2
71d058dec2b52e17bb11cc1a7ef1c967483196342fe1be8228df7a443dd2164c
adam@Adams-Vivobook-Pro:~ $ docker run -itd -p 8041:8042 --net=hadoop --name hadoop-slave2 --hostname hadoop-slave2 liliasfaxi/spark-ha
doop:hv-2.7.2
7c980ec6d5bd250ca41b08eff46fdbe445ad51ebbb75b5f892a96e8e55bea8d4
adam@Adams-Vivobook-Pro:~ $
```

3.6 Starting Hadoop



```
adam@Adams-Vivobook-Pro:~ $ docker exec -it hadoop-master bash
root@hadoop-master:~# ./start-hadoop.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.24.0.2' (ECDSA) to the list of known hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-hadoop-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.24.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.24.0.4' (ECDSA) to the list of known hosts.
hadoop-slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave2.out
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-hadoop-master.out


starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-hadoop-master.out
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.24.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.24.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave2.out
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out


root@hadoop-master:~# |
```

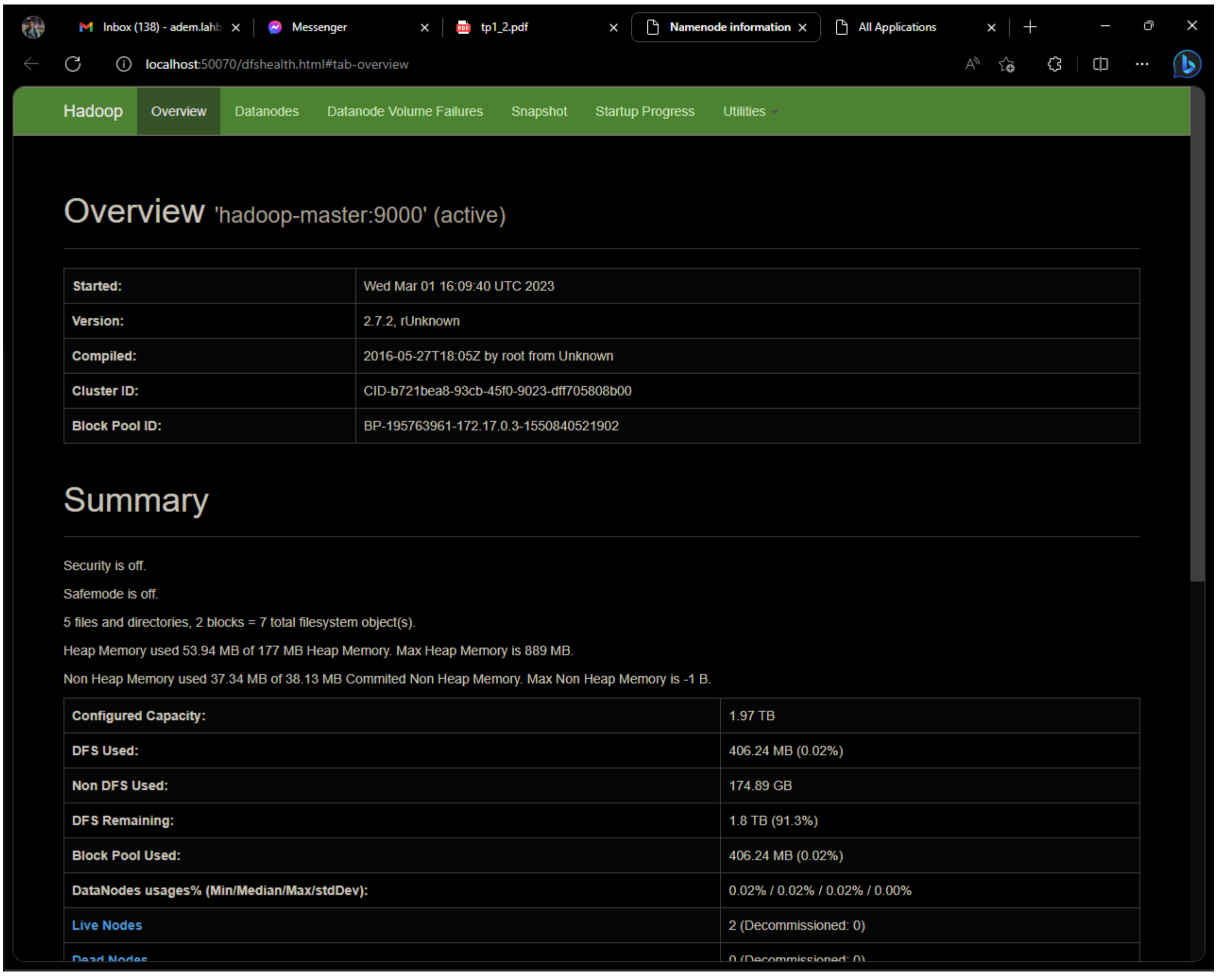
3.7 First Steps with Hadoop



```
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.24.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.24.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave2.out
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out


root@hadoop-master:~# hadoop fs -mkdir -p input
root@hadoop-master:~# hadoop fs -put purchases.txt input
root@hadoop-master:~# hadoop fs -ls input
Found 1 items
-rw-r--r--   2 root supergroup  211312924 2023-03-01 16:11 input/purchases.txt
root@hadoop-master:~# hadoop fs -tail input/purchases.txt
31      17:59   Norfolk Toys    164.34  MasterCard
2012-12-31      17:59   Chula Vista     Music   380.67  Visa
2012-12-31      17:59   Hialeah Toys    115.21  MasterCard
2012-12-31      17:59   Indianapolis    Men's Clothing  158.28  MasterCard
2012-12-31      17:59   Norfolk Garden  414.09  MasterCard
2012-12-31      17:59   Baltimore       DVDs    467.3   Visa
2012-12-31      17:59   Santa Ana       Video Games     144.73  Visa
2012-12-31      17:59   Gilbert Consumer Electronics    354.66  Discover
2012-12-31      17:59   Memphis Sporting Goods  124.79  Amex
2012-12-31      17:59   Chicago Men's Clothing  386.54  MasterCard
2012-12-31      17:59   Birmingham      CDs     118.04  Cash
2012-12-31      17:59   Las Vegas       Health and Beauty       420.46  Amex
2012-12-31      17:59   Wichita Toys    383.9   Cash
2012-12-31      17:59   Tucson  Pet Supplies    268.39  MasterCard
2012-12-31      17:59   Glendale        Women's Clothing        68.05   Amex
2012-12-31      17:59   Albuquerque     Toys    345.7   MasterCard
2012-12-31      17:59   Rochester       DVDs    399.57  Amex
2012-12-31      17:59   Greensboro      Baby    277.27  Discover
2012-12-31      17:59   Arlington       Women's Clothing        134.95  MasterCard
2012-12-31      17:59   Corpus Christi  DVDs    441.61  Discover
root@hadoop-master:~#
```

3.8 Hadoop Web UI

The Hadoop Web UI is a web interface that allows you to monitor the status of your Hadoop cluster. It is available at `http://localhost:50070/`. To view job results and progress, go to `http://localhost:8088/`.



Overview 'hadoop-master:9000' (active)

Started:	Wed Mar 01 16:09:40 UTC 2023
Version:	2.7.2, rUnknown
Compiled:	2016-05-27T18:05Z by root from Unknown
Cluster ID:	CID-b721bea8-93cb-45f0-9023-dff705808b00
Block Pool ID:	BP-195763961-172.17.0.3-1550840521902

Summary

Security is off.

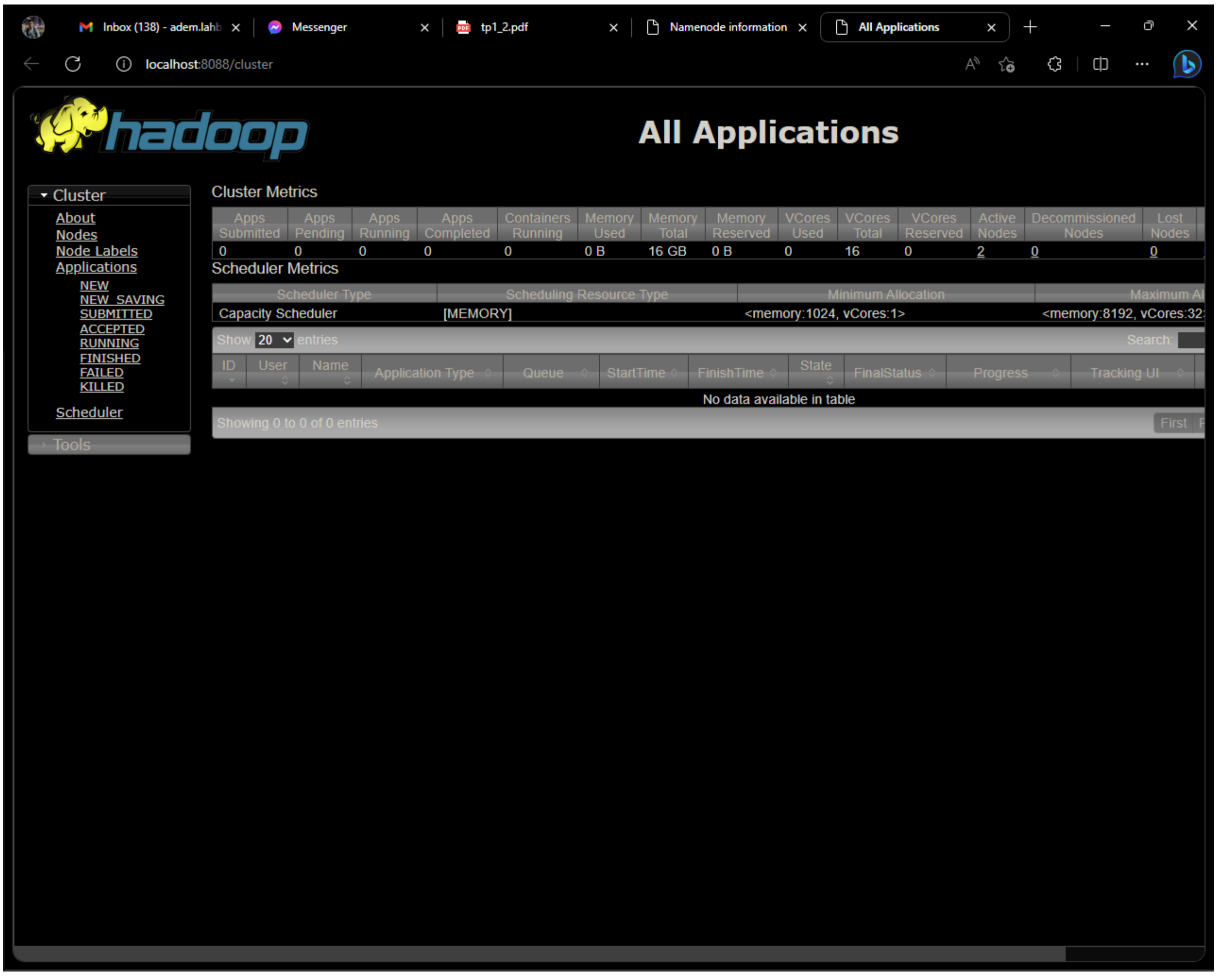
Safemode is off.

5 files and directories, 2 blocks = 7 total filesystem object(s).

Heap Memory used 53.94 MB of 177 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 37.34 MB of 38.13 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	1.97 TB
DFS Used:	406.24 MB (0.02%)
Non DFS Used:	174.89 GB
DFS Remaining:	1.8 TB (91.3%)
Block Pool Used:	406.24 MB (0.02%)
DataNodes usages% (Min/Median/Max/stdDev):	0.02% / 0.02% / 0.02% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes
0	0	0	0	0	0 B	16 GB	0 B	0	16	0	2	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Al
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

Showing 0 to 0 of 0 entries First

4 Application

4.1 Idea

Total sales per shop is a simple MapReduce application that counts the total sales per shop. The input is a CSV file with the architecure: date, time, shop, product, price, payment method. The output is a CSV file with the architecture: shop, total sales.

4.2 Code Modifications

4.2.1 WordCount.java

```java
package tn.insat.tp1;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(FloatWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

4.2.2 TokenizerMapper.java

The TokenizerMapper class extends the Mapper class and overrides the map method. The map method takes three parameters: the key, the value, and the Context. The key is the offset of the line in the input file, the value is the line itself, and the Context is the object that allows you to write the output of the map method. The map method is called once for each line in the input file. The map method tokenizes the line into words and writes each word as a key with a value of 1. The output of the map method is written to the Context object.

```java
package tn.insat.tp1;

import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;
import java.util.StringTokenizer;

public class TokenizerMapper
        extends Mapper<Object, Text, Text, FloatWritable>{

    private Text magasin = new Text();
    private FloatWritable cout = new FloatWritable();

    public void map(Object key, Text value, Mapper.Context context) throws IOException,
    ↪   InterruptedException {

        String[] columns = value.toString().split(",");

        if (columns.length == 7) {
            // Extract the 'magasin' and 'cout' values
            magasin.set(columns[3]);
            cout.set(Float.parseFloat(columns[5]));
            System.out.println(magasin);
            System.out.println(cout);
            context.write(magasin, cout);
        }
    }
}
```

4.2.3 IntSumReducer.java

The IntSumReducer class extends the Reducer class and overrides the reduce method. The reduce method takes three parameters: the key, the values, and the Context. The key is the word, the values are the counts, and the Context is the object that allows you to write the output of the reduce method.

The reduce method is called once for each key. The reduce method sums the values for each key and writes the key and the sum as the output. The output of the reduce method is written to the Context object.

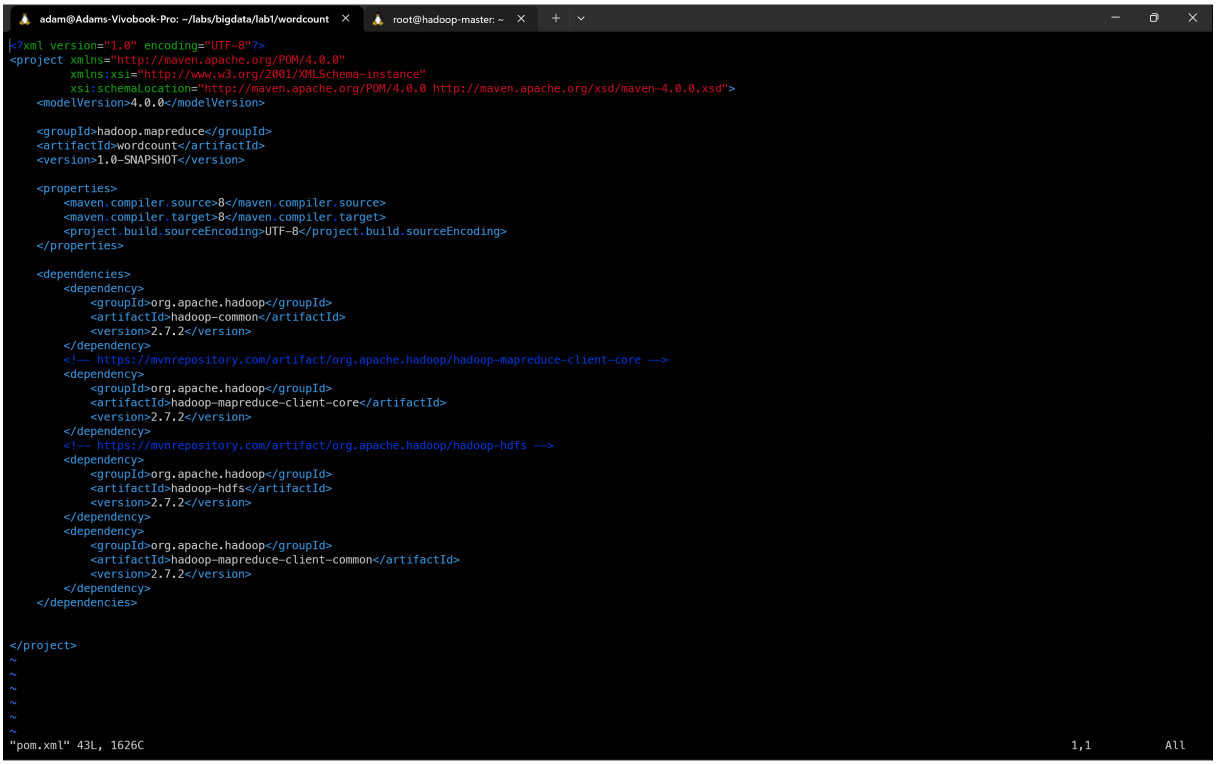
```java
package tn.insat.tp1;


import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class IntSumReducer
        extends Reducer<Text,FloatWritable,Text,FloatWritable> {

    private FloatWritable result = new FloatWritable();

    public void reduce(Text key, Iterable<FloatWritable> values,
                       Context context
    ) throws IOException, InterruptedException {
        int sum = 0;
        for (FloatWritable val : values) {
            System.out.println("value: "+val.get());
            sum += val.get();
        }
        System.out.println("--> Sum = "+sum);
        result.set(sum);
        context.write(key, result);
    }
}
```

Since prices are rather floats, I used FloatWritable instead of IntWritable.

4.2.4 Github repo for source code: https://github.com/adamlahbib/BigData_Labs/tree/main/lab1/wordcount

4.2.5 `pom.xml` File

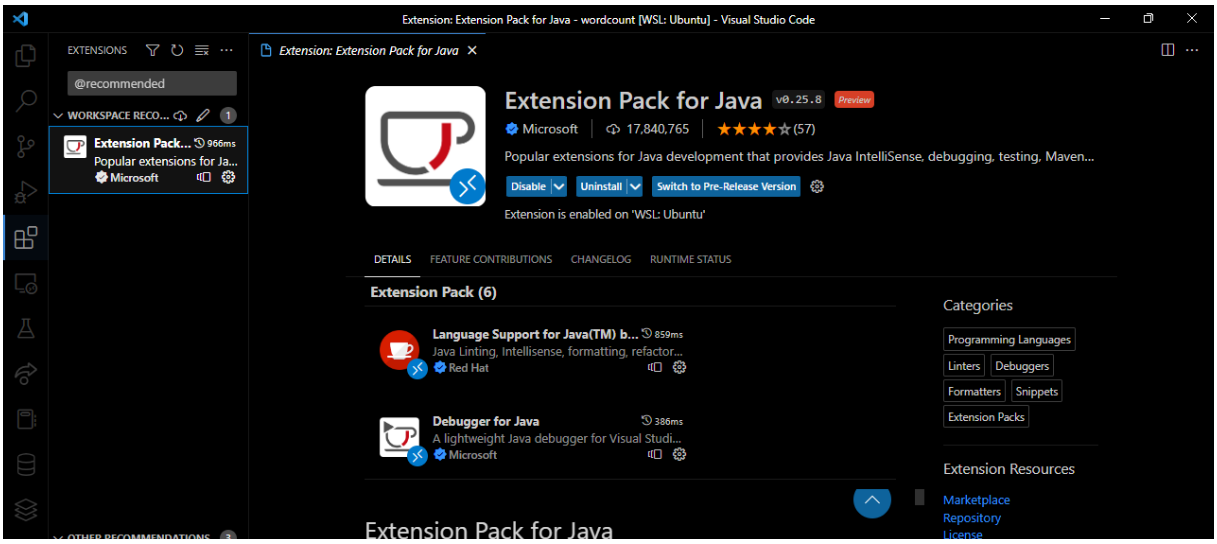
This is the pom.xml file, it contains all the dependencies needed for the project, I used the following dependencies:



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>hadoop.mapreduce</groupId>
    <artifactId>wordcount</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-common</artifactId>
            <version>2.7.2</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-client-core -->
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-mapreduce-client-core</artifactId>
            <version>2.7.2</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-hdfs -->
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
            <version>2.7.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-mapreduce-client-common</artifactId>
            <version>2.7.2</version>
        </dependency>
    </dependencies>


</project>
~
~
~
~
~
~
"pom.xml" 43L, 1626C                                                        1,1           All
```

4.3 Local test using VSCode

I will use the following extension to run/debug Java code in VSCode:

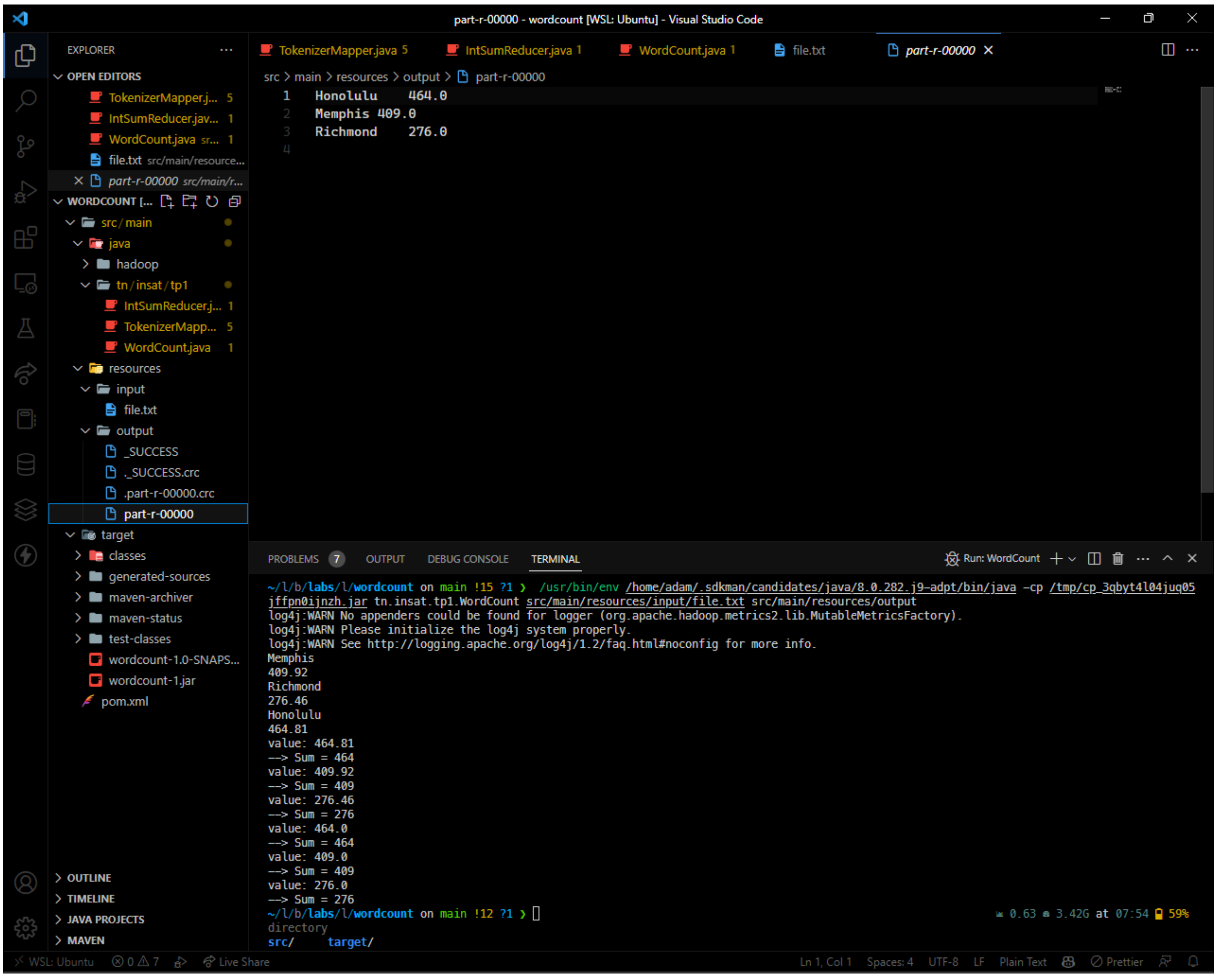


Extension: Extension Pack for Java - wordcount [WSL: Ubuntu] - Visual Studio Code
EXTENSIONS
@recommended
WORKSPACE RECO...
1
Extension Pack... 966ms
Popular extensions for Ja...
Microsoft
OTHER RECOMMENDATIONS
Extension: Extension Pack for Java
Extension Pack for Java
v0.25.8
Preview
Microsoft
17,840,765
★★★★☆ (57)
Popular extensions for Java development that provides Java IntelliSense, debugging, testing, Maven...
Disable
Uninstall
Switch to Pre-Release Version
Extension is enabled on 'WSL: Ubuntu'
DETAILS
FEATURE CONTRIBUTIONS
CHANGELOG
RUNTIME STATUS
Extension Pack (6)
Language Support for Java(TM) b... 859ms
Java Linting, Intellisense, formatting, refactor...
Red Hat
Debugger for Java 386ms
A lightweight Java debugger for Visual Studi...
Microsoft
Extension Pack for Java
Categories
Programming Languages
Linters
Debuggers
Formatters
Snippets
Extension Packs
Extension Resources
Marketplace
Repository
License

I will use this part of the file as my input:

```
428400,2012-02-07,16:17,Memphis,Crafts,409.92,Discover
428401,2012-02-07,16:17,Richmond,Children's Clothing,276.46,MasterCard
428402,2012-02-07,16:17,Honolulu,Crafts,464.81,MasterCard
```

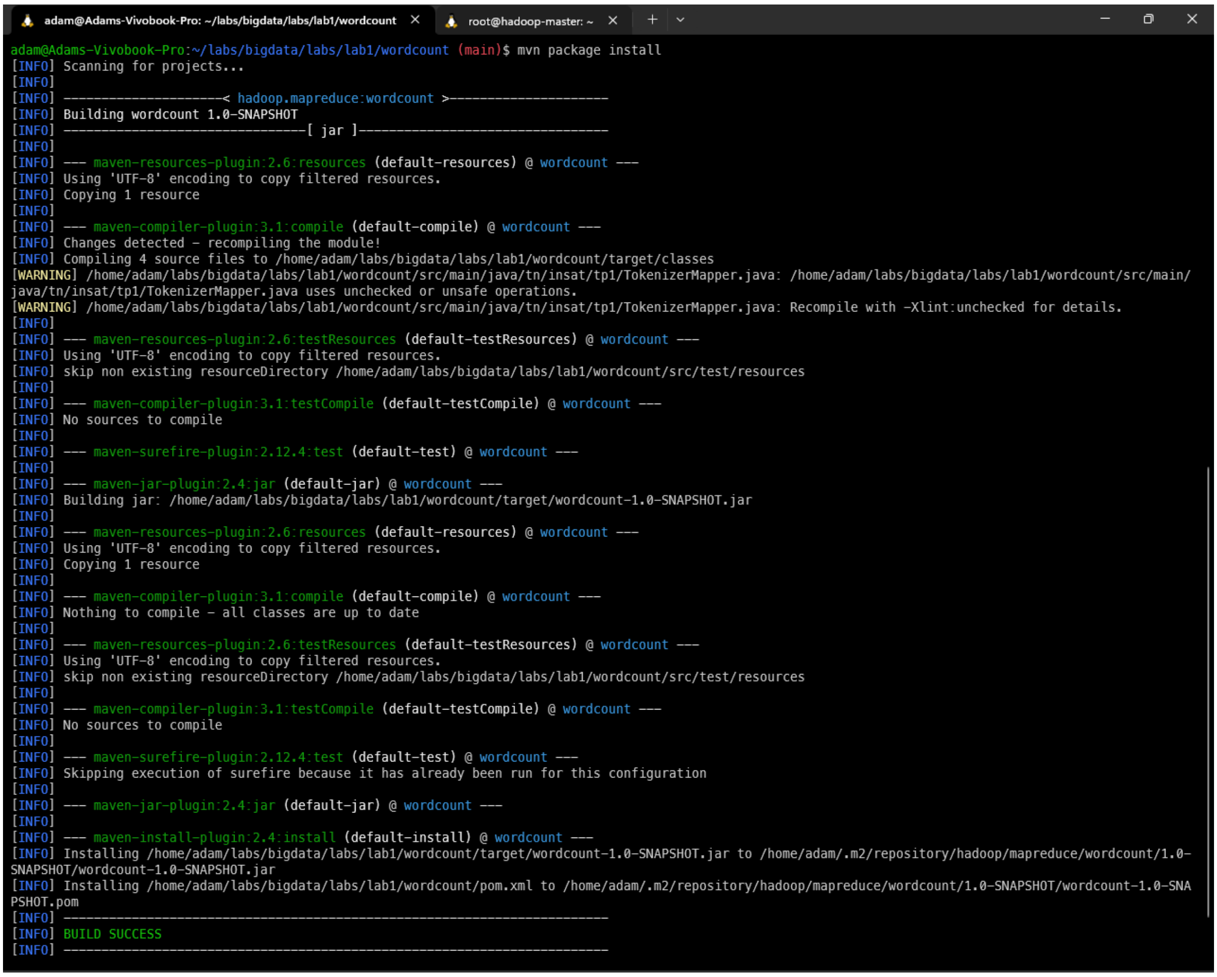
Then thanks to the VSCode extension when I press RUN and add the arguments to the command line, I get the following output:



```
Honolulu	464.0
Memphis	409.0
Richmond	276.0
```

```
~/l/b/labs/l/wordcount on main !15 ?1 ❯  /usr/bin/env /home/adam/.sdkman/candidates/java/8.0.282.j9-adpt/bin/java -cp /tmp/cp_3qbyt4l04juq05
jffpn0ijnzh.jar tn.insat.tp1.WordCount src/main/resources/input/file.txt src/main/resources/output
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Memphis
409.92
Richmond
276.46
Honolulu
464.81
value: 464.81
---> Sum = 464
value: 409.92
---> Sum = 409
value: 276.46
---> Sum = 276
value: 464.0
---> Sum = 464
value: 409.0
---> Sum = 409
value: 276.0
---> Sum = 276
~/l/b/labs/l/wordcount on main !12 ?1 ❯                                    0.63  3.42G at 07:54  59%
directory
src/     target/
```

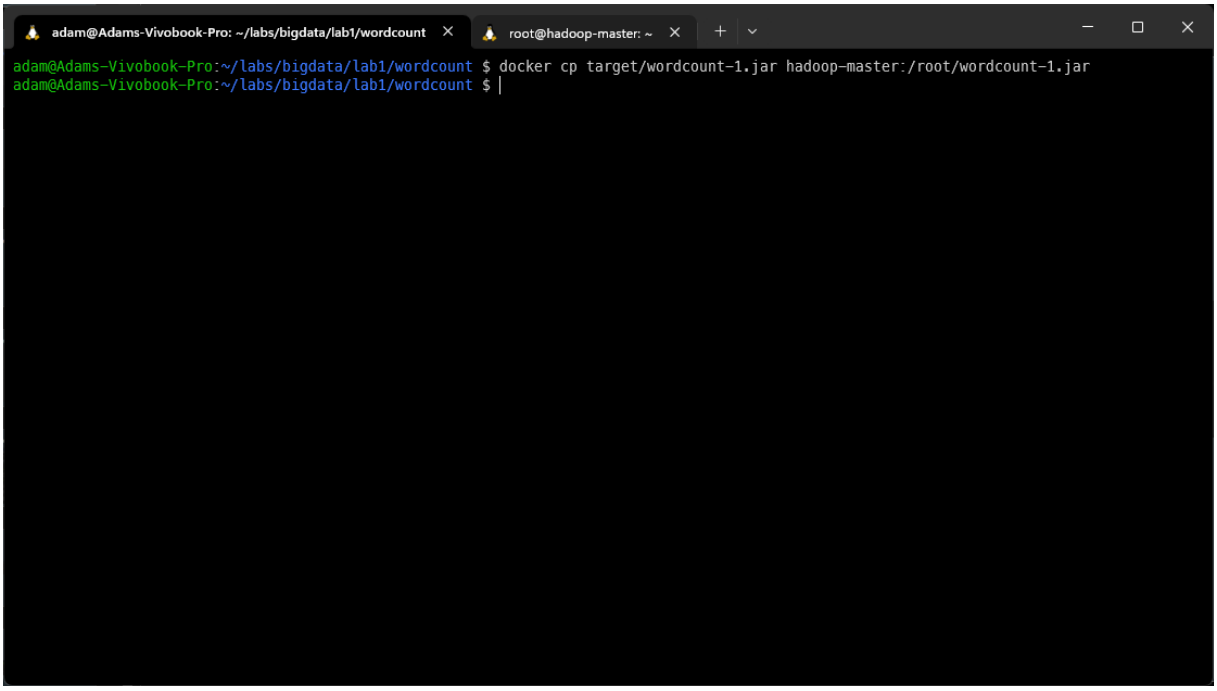
4.3.1 Packages Installation: `mvn package install`



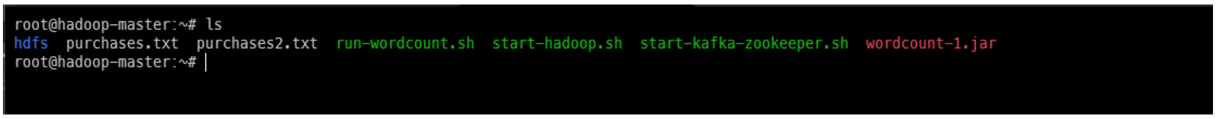
```
adam@Adams-Vivobook-Pro:~/labs/bigdata/labs/lab1/wordcount (main)$ mvn package install
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< hadoop.mapreduce:wordcount >--------------------
[INFO] Building wordcount 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ wordcount ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ wordcount ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 4 source files to /home/adam/labs/bigdata/labs/lab1/wordcount/target/classes
[WARNING] /home/adam/labs/bigdata/labs/lab1/wordcount/src/main/java/tn/insat/tp1/TokenizerMapper.java: /home/adam/labs/bigdata/labs/lab1/wordcount/src/main/
java/tn/insat/tp1/TokenizerMapper.java uses unchecked or unsafe operations.
[WARNING] /home/adam/labs/bigdata/labs/lab1/wordcount/src/main/java/tn/insat/tp1/TokenizerMapper.java: Recompile with -Xlint:unchecked for details.
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ wordcount ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/adam/labs/bigdata/labs/lab1/wordcount/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ wordcount ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ wordcount ---
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ wordcount ---
[INFO] Building jar: /home/adam/labs/bigdata/labs/lab1/wordcount/target/wordcount-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ wordcount ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ wordcount ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ wordcount ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/adam/labs/bigdata/labs/lab1/wordcount/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ wordcount ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ wordcount ---
[INFO] Skipping execution of surefire because it has already been run for this configuration
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ wordcount ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ wordcount ---
[INFO] Installing /home/adam/labs/bigdata/labs/lab1/wordcount/target/wordcount-1.0-SNAPSHOT.jar to /home/adam/.m2/repository/hadoop/mapreduce/wordcount/1.0-
SNAPSHOT/wordcount-1.0-SNAPSHOT.jar
[INFO] Installing /home/adam/labs/bigdata/labs/lab1/wordcount/pom.xml to /home/adam/.m2/repository/hadoop/mapreduce/wordcount/1.0-SNAPSHOT/wordcount-1.0-SNA
PSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

4.4 Hadoop test (Dockerized)

For hadoop test, I will use the CSV file as input, instead of the tab delimited file, to make things easier.

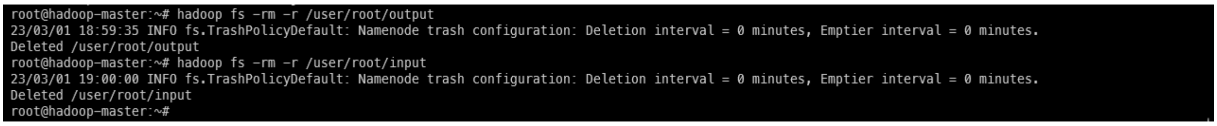


```
adam@Adams-Vivobook-Pro:~/labs/bigdata/lab1/wordcount $ docker cp target/wordcount-1.jar hadoop-master:/root/wordcount-1.jar
adam@Adams-Vivobook-Pro:~/labs/bigdata/lab1/wordcount $ |
```



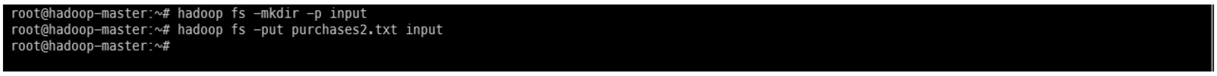
```
root@hadoop-master:~# ls
hdfs  purchases.txt  purchases2.txt  run-wordcount.sh  start-hadoop.sh  start-kafka-zookeeper.sh  wordcount-1.jar
root@hadoop-master:~# |
```

Let’s remove the old input content as well as the output folder



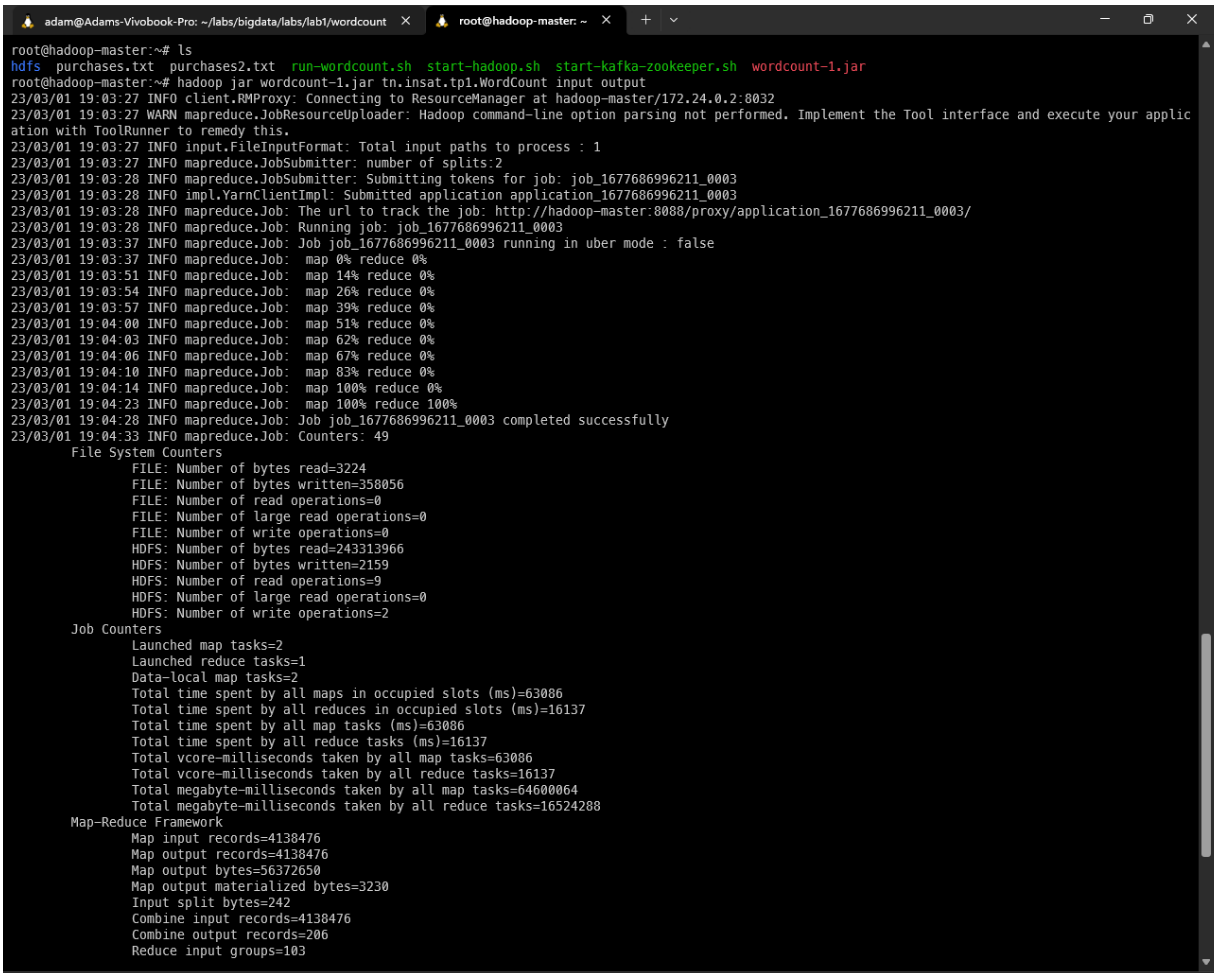
```
root@hadoop-master:~# hadoop fs -rm -r /user/root/output
23/03/01 18:59:35 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/root/output
root@hadoop-master:~# hadoop fs -rm -r /user/root/input
23/03/01 19:00:00 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/root/input
root@hadoop-master:~#
```

Let’s now use purchases2.txt

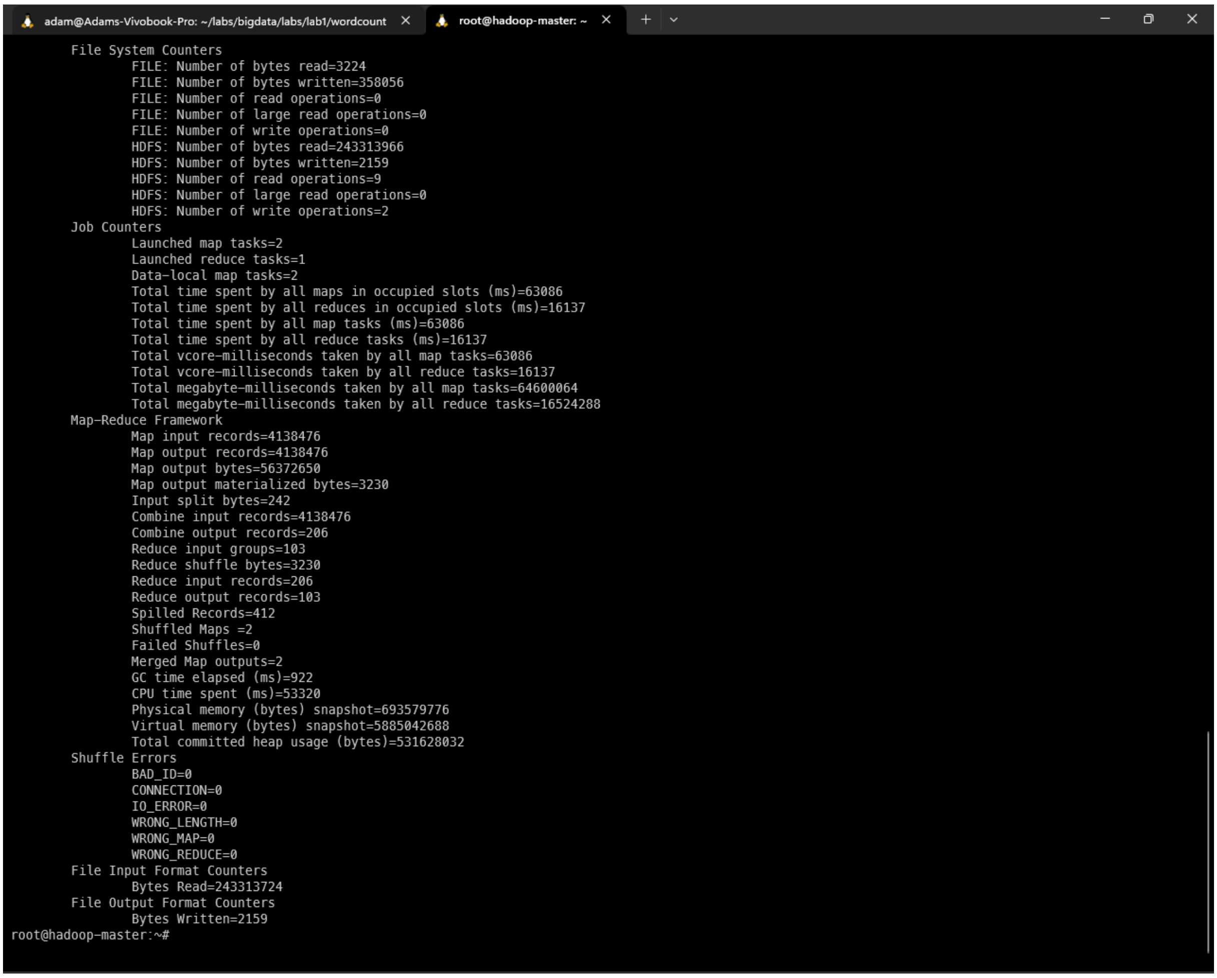


```
root@hadoop-master:~# hadoop fs -mkdir -p input
root@hadoop-master:~# hadoop fs -put purchases2.txt input
root@hadoop-master:~#
```

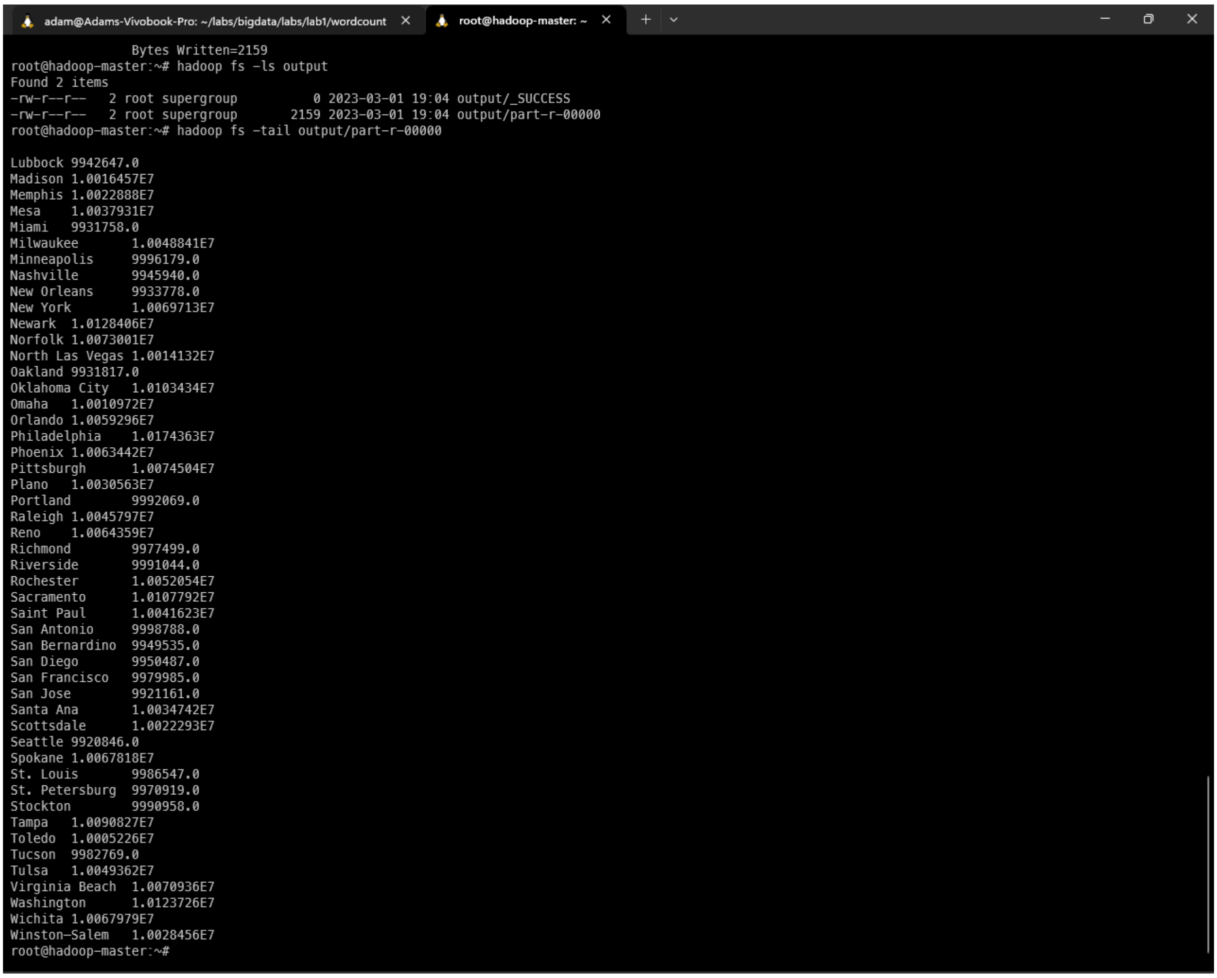
Everything is ready, let’s run the job



```
root@hadoop-master:~# ls
hdfs  purchases.txt  purchases2.txt  run-wordcount.sh  start-hadoop.sh  start-kafka-zookeeper.sh  wordcount-1.jar
root@hadoop-master:~# hadoop jar wordcount-1.jar tn.insat.tp1.WordCount input output
23/03/01 19:03:27 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.24.0.2:8032
23/03/01 19:03:27 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your applic
ation with ToolRunner to remedy this.
23/03/01 19:03:27 INFO input.FileInputFormat: Total input paths to process : 1
23/03/01 19:03:27 INFO mapreduce.JobSubmitter: number of splits:2
23/03/01 19:03:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1677686996211_0003
23/03/01 19:03:28 INFO impl.YarnClientImpl: Submitted application application_1677686996211_0003
23/03/01 19:03:28 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1677686996211_0003/
23/03/01 19:03:28 INFO mapreduce.Job: Running job: job_1677686996211_0003
23/03/01 19:03:37 INFO mapreduce.Job: Job job_1677686996211_0003 running in uber mode : false
23/03/01 19:03:37 INFO mapreduce.Job:  map 0% reduce 0%
23/03/01 19:03:51 INFO mapreduce.Job:  map 14% reduce 0%
23/03/01 19:03:54 INFO mapreduce.Job:  map 26% reduce 0%
23/03/01 19:03:57 INFO mapreduce.Job:  map 39% reduce 0%
23/03/01 19:04:00 INFO mapreduce.Job:  map 51% reduce 0%
23/03/01 19:04:03 INFO mapreduce.Job:  map 62% reduce 0%
23/03/01 19:04:06 INFO mapreduce.Job:  map 67% reduce 0%
23/03/01 19:04:10 INFO mapreduce.Job:  map 83% reduce 0%
23/03/01 19:04:14 INFO mapreduce.Job:  map 100% reduce 0%
23/03/01 19:04:23 INFO mapreduce.Job:  map 100% reduce 100%
23/03/01 19:04:28 INFO mapreduce.Job: Job job_1677686996211_0003 completed successfully
23/03/01 19:04:33 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=3224
                FILE: Number of bytes written=358056
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=243313966
                HDFS: Number of bytes written=2159
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=63086
                Total time spent by all reduces in occupied slots (ms)=16137
                Total time spent by all map tasks (ms)=63086
                Total time spent by all reduce tasks (ms)=16137
                Total vcore-milliseconds taken by all map tasks=63086
                Total vcore-milliseconds taken by all reduce tasks=16137
                Total megabyte-milliseconds taken by all map tasks=64600064
                Total megabyte-milliseconds taken by all reduce tasks=16524288
        Map-Reduce Framework
                Map input records=4138476
                Map output records=4138476
                Map output bytes=56372650
                Map output materialized bytes=3230
                Input split bytes=242
                Combine input records=4138476
                Combine output records=206
                Reduce input groups=103
```



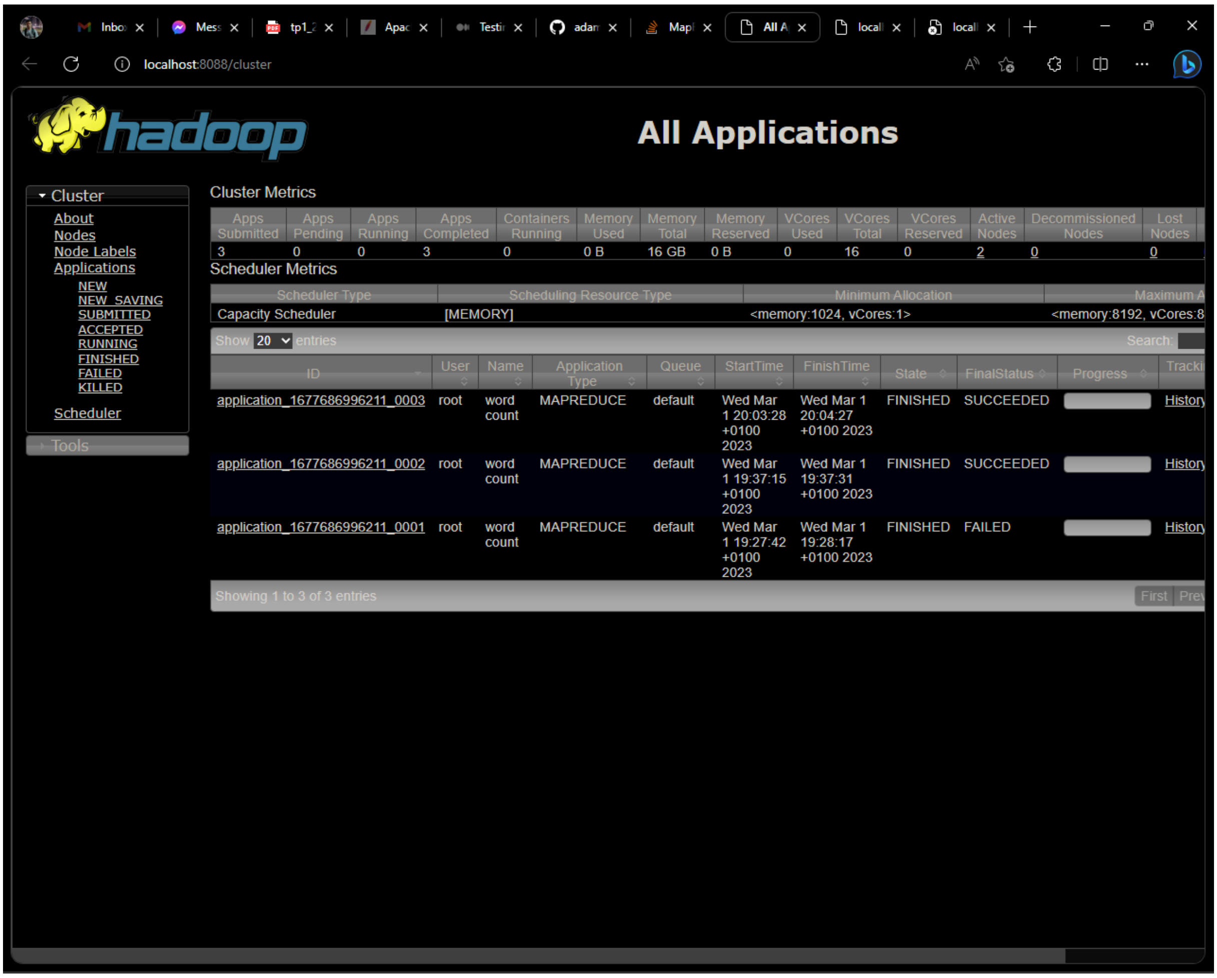
```
	File System Counters
		FILE: Number of bytes read=3224
		FILE: Number of bytes written=358056
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=243313966
		HDFS: Number of bytes written=2159
		HDFS: Number of read operations=9
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=2
		Launched reduce tasks=1
		Data-local map tasks=2
		Total time spent by all maps in occupied slots (ms)=63086
		Total time spent by all reduces in occupied slots (ms)=16137
		Total time spent by all map tasks (ms)=63086
		Total time spent by all reduce tasks (ms)=16137
		Total vcore-milliseconds taken by all map tasks=63086
		Total vcore-milliseconds taken by all reduce tasks=16137
		Total megabyte-milliseconds taken by all map tasks=64600064
		Total megabyte-milliseconds taken by all reduce tasks=16524288
	Map-Reduce Framework
		Map input records=4138476
		Map output records=4138476
		Map output bytes=56372650
		Map output materialized bytes=3230
		Input split bytes=242
		Combine input records=4138476
		Combine output records=206
		Reduce input groups=103
		Reduce shuffle bytes=3230
		Reduce input records=206
		Reduce output records=103
		Spilled Records=412
		Shuffled Maps =2
		Failed Shuffles=0
		Merged Map outputs=2
		GC time elapsed (ms)=922
		CPU time spent (ms)=53320
		Physical memory (bytes) snapshot=693579776
		Virtual memory (bytes) snapshot=5885042688
		Total committed heap usage (bytes)=531628032
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=243313724
	File Output Format Counters
		Bytes Written=2159
root@hadoop-master:~#
```



```
                Bytes Written=2159
root@hadoop-master:~# hadoop fs -ls output
Found 2 items
-rw-r--r--   2 root supergroup          0 2023-03-01 19:04 output/_SUCCESS
-rw-r--r--   2 root supergroup       2159 2023-03-01 19:04 output/part-r-00000
root@hadoop-master:~# hadoop fs -tail output/part-r-00000

Lubbock 9942647.0
Madison 1.0016457E7
Memphis 1.0022888E7
Mesa    1.0037931E7
Miami   9931758.0
Milwaukee       1.0048841E7
Minneapolis     9996179.0
Nashville       9945940.0
New Orleans     9933778.0
New York        1.0069713E7
Newark  1.0128406E7
Norfolk 1.0073001E7
North Las Vegas 1.0014132E7
Oakland 9931817.0
Oklahoma City   1.0103434E7
Omaha   1.0010972E7
Orlando 1.0059296E7
Philadelphia    1.0174363E7
Phoenix 1.0063442E7
Pittsburgh      1.0074504E7
Plano   1.0030563E7
Portland        9992069.0
Raleigh 1.0045797E7
Reno    1.0064359E7
Richmond        9977499.0
Riverside       9991044.0
Rochester       1.0052054E7
Sacramento      1.0107792E7
Saint Paul      1.0041623E7
San Antonio     9998788.0
San Bernardino  9949535.0
San Diego       9950487.0
San Francisco   9979985.0
San Jose        9921161.0
Santa Ana       1.0034742E7
Scottsdale      1.0022293E7
Seattle 9920846.0
Spokane 1.0067818E7
St. Louis       9986547.0
St. Petersburg  9970919.0
Stockton        9990958.0
Tampa   1.0090827E7
Toledo  1.0005226E7
Tucson  9982769.0
Tulsa   1.0049362E7
Virginia Beach  1.0070936E7
Washington      1.0123726E7
Wichita 1.0067979E7
Winston-Salem   1.0028456E7
root@hadoop-master:~#
```

4.5 Monitoring the job

Monitoring the job is done using the Hadoop Web UI, we can see the job running on the Master node:



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes
3	0	0	3	0	0 B	16 GB	0 B	0	16	0	2	0	0

Scheduler Metrics

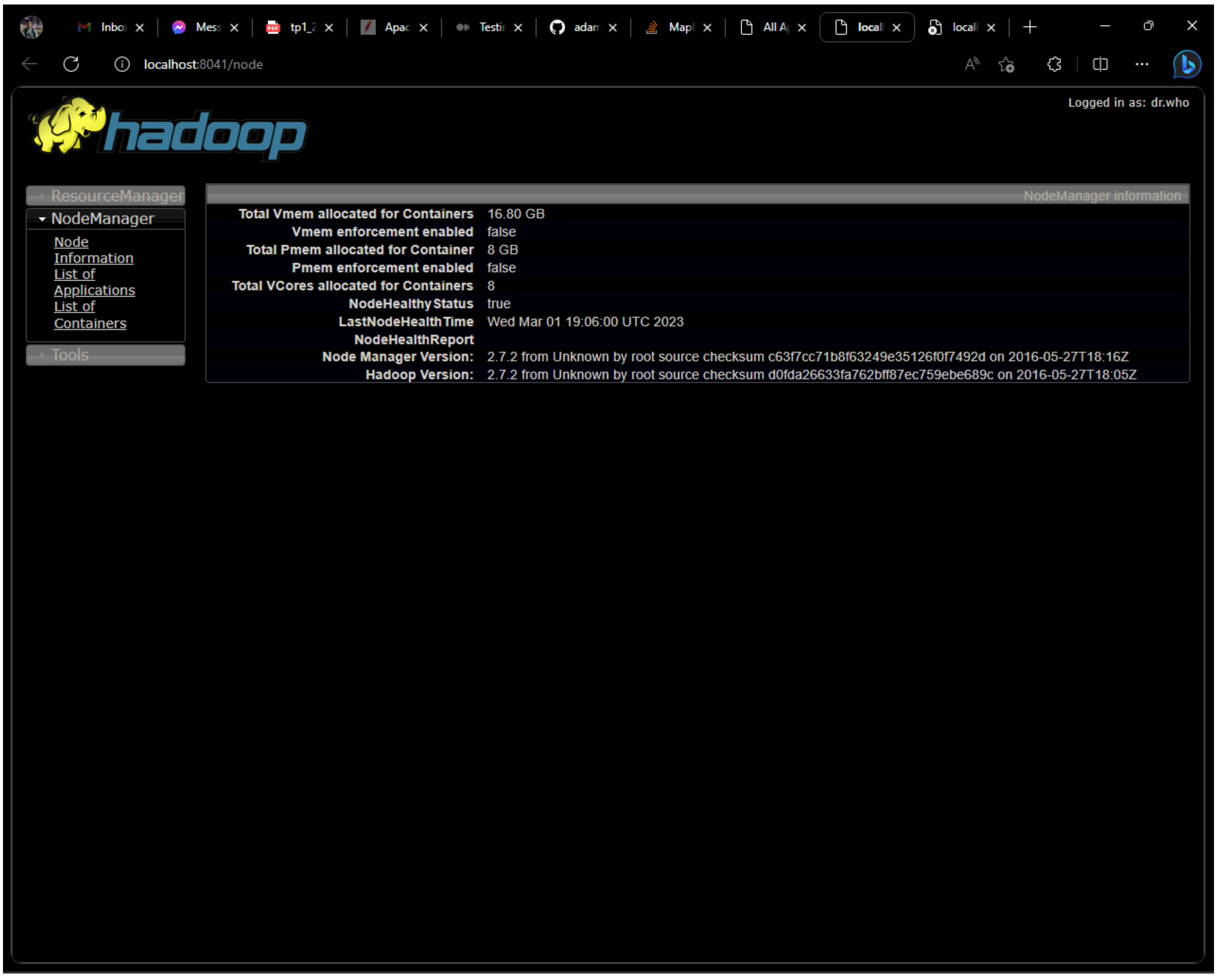
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum A
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracki
application_1677686996211_0003	root	word count	MAPREDUCE	default	Wed Mar 1 20:03:28 +0100 2023	Wed Mar 1 20:04:27 +0100 2023	FINISHED	SUCCEEDED		History
application_1677686996211_0002	root	word count	MAPREDUCE	default	Wed Mar 1 19:37:15 +0100 2023	Wed Mar 1 19:37:31 +0100 2023	FINISHED	SUCCEEDED		History
application_1677686996211_0001	root	word count	MAPREDUCE	default	Wed Mar 1 19:27:42 +0100 2023	Wed Mar 1 19:28:17 +0100 2023	FINISHED	FAILED		History

Showing 1 to 3 of 3 entries

and Slave 1 for example:



NodeManager information

Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	false
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	false
Total VCores allocated for Containers	8
NodeHealthy Status	true
LastNodeHealth Time	Wed Mar 01 19:06:00 UTC 2023
NodeHealthReport	
Node Manager Version:	2.7.2 from Unknown by root source checksum c63f7cc71b8f63249e35126f0f7492d on 2016-05-27T18:16Z
Hadoop Version:	2.7.2 from Unknown by root source checksum d0fda26633fa762bff87ec759ebe689c on 2016-05-27T18:05Z

5 Homework

5.1 Idea

Processing Wireshark PCAP files using Hadoop MapReduce job, to extract the GEO data from PCAP packets using the MaxMind GeoLite2 database.

5.2 Architecture

5.3 Code

We will, as usual, modify the WordCount (template) example to process the PCAP files.

5.3.1 `WordCount.java`

```
package tn.insat.tp1;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.SequenceFileAsBinaryInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class Driver extends Configured implements Tool {

    public int run(String[] args) throws Exception {

        Job job = Job.getInstance(getConf(), "PcapProcessing");
        job.setJarByClass(getClass());

        Path in = new Path(args[0]);
        Path out = new Path(args[1]);
        FileInputFormat.setInputPaths(job, in);
        FileOutputFormat.setOutputPath(job, out);

        job.setMapperClass(PcapMapper.class);
        job.setReducerClass(PcapReducer.class);

        job.setInputFormatClass(SequenceFileAsBinaryInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        job.addCacheFile(new Path("GeoLite2-City.mmdb").toUri());

        return job.waitForCompletion(true)?0:1;
    }

    public static void main(String[] args) throws Exception {
        int result = ToolRunner.run(new Configuration(),new Driver(), args);
                System.exit(result);

    }

}
```

5.3.2 `TokenizerMapper.java`

```
package tn.insat.tp1;

import java.io.IOException;
import java.io.InputStream;
import java.net.InetAddress;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.BytesWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import com.maxmind.geoip2.DatabaseReader;
```

```java
import com.maxmind.geoip2.exception.GeoIp2Exception;
import com.maxmind.geoip2.model.CityResponse;

public class PcapMapper extends Mapper<BytesWritable, BytesWritable, Text, IntWritable> {


 private static final String DBNAME="GeoLite2-City.mmdb";

 private Text word = new Text();
 private final static IntWritable one = new IntWritable(1);
 private byte[] binaryValue;
 private byte[] binaryIpAddress = new byte[4];
 private InputStream is;
 private DatabaseReader dbReader;
 private CityResponse dbResponse;

 @Override
protected void setup(Context context) throws IOException, InterruptedException {
     Configuration conf = context.getConfiguration();

     FileSystem fs = FileSystem.getLocal(conf);
     is = fs.open(new Path(DBNAME));
     dbReader = new DatabaseReader.Builder(is).build();
}

@Override
 public void map(BytesWritable key, BytesWritable value, Context contex) throws IOException, InterruptedException {

     // Get Bytes
     binaryValue = value.getBytes();

     // Extract IP address
     binaryIpAddress[0] = binaryValue[27];
     binaryIpAddress[1] = binaryValue[28];
     binaryIpAddress[2] = binaryValue[29];
     binaryIpAddress[3] = binaryValue[30];

     // Find GEO data of the IP address
     try {
        dbResponse = dbReader.city(InetAddress.getByAddress(binaryIpAddress));
        word.set(dbResponse.getCountry().getName()+"-"+dbResponse.getCity().getName());
        contex.write(word, one);
    } catch (GeoIp2Exception e) {
        e.printStackTrace();
    }
 }

@Override
protected void cleanup(Context context) throws IOException, InterruptedException {
    is.close();
}
}
```

We know that in a pcap file, the Source IP address is stored in the 27th to 30th bytes of the packet, so we extract it and use it to find the GEO data.

5.3.3 `IntSumReducer.java`

```java
package tn.insat.tp1;

import java.io.IOException;
import java.util.Iterator;


import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;



public class PcapReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

 private IntWritable totalWordCount = new IntWritable();

 @Override
 public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
 ↪  IOException, InterruptedException {

  int wordCount = 0;

  Iterator<IntWritable> it=values.iterator();
  while (it.hasNext()) {
   wordCount += it.next().get();
  }

  // Write keys with count > 1
  if(wordCount > 1){
      totalWordCount.set(wordCount);
      context.write(key, totalWordCount);
  }
 }
}
```

You can view the full code at: https://github.com/adamlahbib/BigData_Labs/tree/main/homework

I will use a dummy PCAP file from Wireshark samples as input.

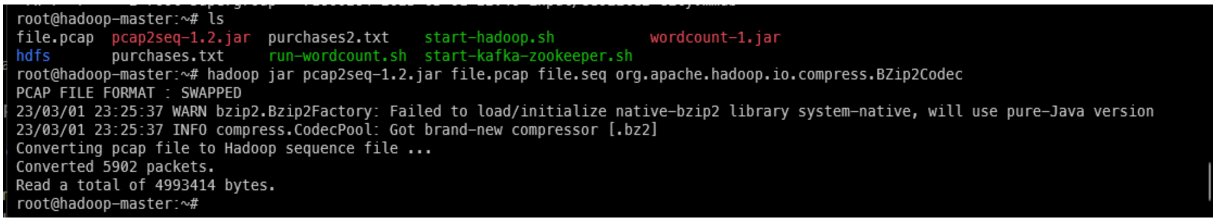
Link: https://tranalyzer.com/download/data/faf-exercise.pcap *I will rename it to file.pcap simply.*

We will need to convert the pcap file to a sequence file using the following tool: https://github.com/marouni/pcap2seq

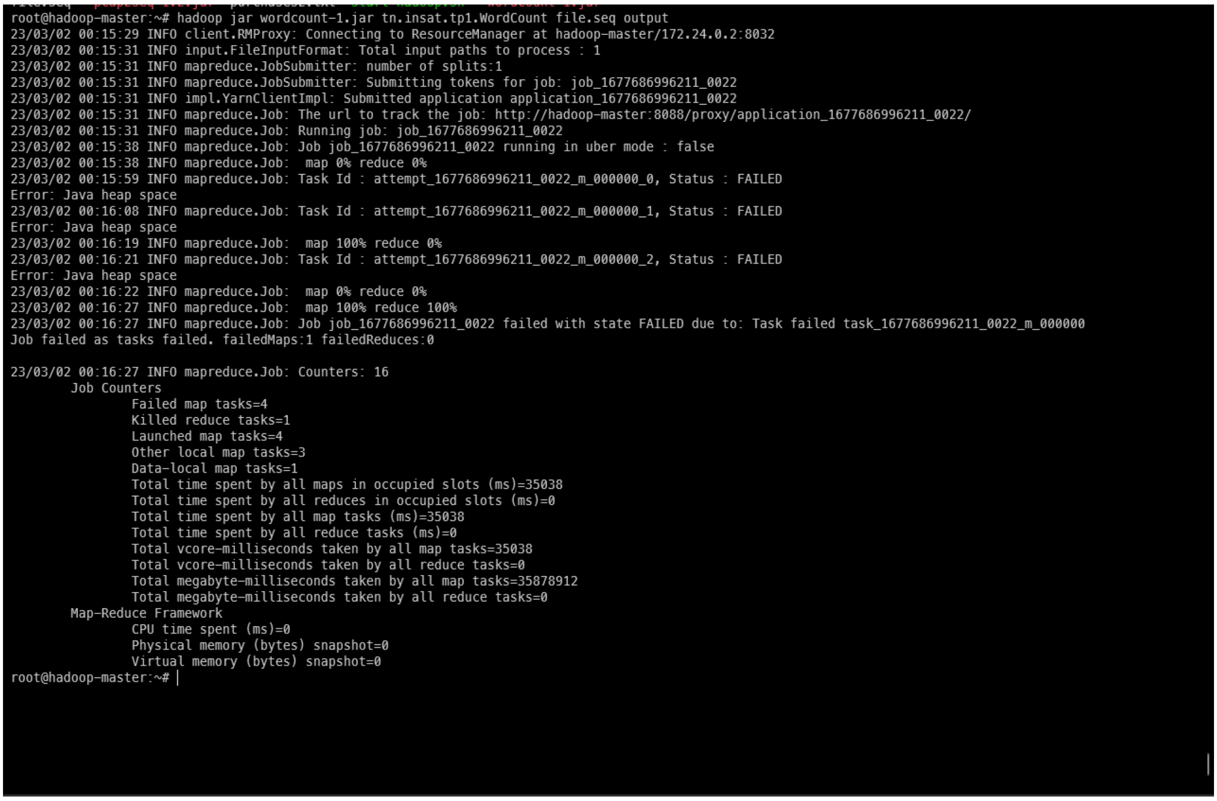
I will also need this DB file to get the GEO data of the IP addresses. https://github.com/P3TERX/GeoLite.mmdb/raw/download/GeoLite2-City.mmdb it weights ~ 60 MB and I wget it to my home directory.

5.4 Results

To conduct a test on the cluster, I had to put the DB file `GeoLite2-City.mmdb` in HDFS, the `pcap2seq.jar` file along the **fat jar** `wordocunt-1.jar` (meaning a jar packed with dependencies) in the hadoop root, and the `file.seq` file in the root of the HDFS (according to the tool!).



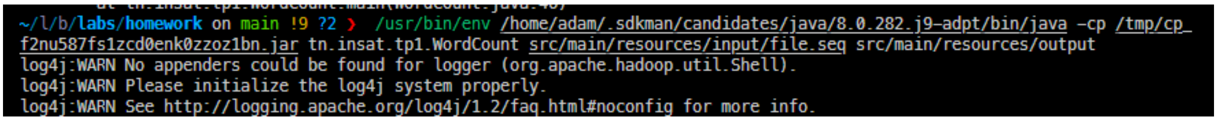
```
root@hadoop-master:~# ls
file.pcap  pcap2seq-1.2.jar  purchases2.txt    start-hadoop.sh           wordcount-1.jar
hdfs       purchases.txt     run-wordcount.sh  start-kafka-zookeeper.sh
root@hadoop-master:~# hadoop jar pcap2seq-1.2.jar file.pcap file.seq org.apache.hadoop.io.compress.BZip2Codec
PCAP FILE FORMAT : SWAPPED
23/03/01 23:25:37 WARN bzip2.Bzip2Factory: Failed to load/initialize native-bzip2 library system-native, will use pure-Java version
23/03/01 23:25:37 INFO compress.CodecPool: Got brand-new compressor [.bz2]
Converting pcap file to Hadoop sequence file ...
Converted 5902 packets.
Read a total of 4993414 bytes.
root@hadoop-master:~#
```



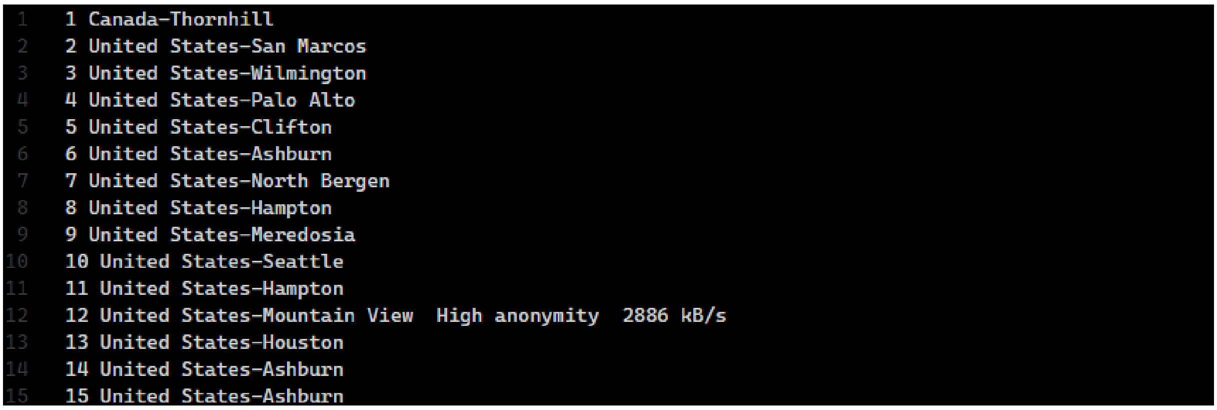
```
root@hadoop-master:~# hadoop jar wordcount-1.jar tn.insat.tp1.WordCount file.seq output
23/03/02 00:15:29 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.24.0.2:8032
23/03/02 00:15:31 INFO input.FileInputFormat: Total input paths to process : 1
23/03/02 00:15:31 INFO mapreduce.JobSubmitter: number of splits:1
23/03/02 00:15:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1677686996211_0022
23/03/02 00:15:31 INFO impl.YarnClientImpl: Submitted application application_1677686996211_0022
23/03/02 00:15:31 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1677686996211_0022/
23/03/02 00:15:31 INFO mapreduce.Job: Running job: job_1677686996211_0022
23/03/02 00:15:38 INFO mapreduce.Job: Job job_1677686996211_0022 running in uber mode : false
23/03/02 00:15:38 INFO mapreduce.Job:  map 0% reduce 0%
23/03/02 00:15:59 INFO mapreduce.Job: Task Id : attempt_1677686996211_0022_m_000000_0, Status : FAILED
Error: Java heap space
23/03/02 00:16:08 INFO mapreduce.Job: Task Id : attempt_1677686996211_0022_m_000000_1, Status : FAILED
Error: Java heap space
23/03/02 00:16:19 INFO mapreduce.Job:  map 100% reduce 0%
23/03/02 00:16:21 INFO mapreduce.Job: Task Id : attempt_1677686996211_0022_m_000000_2, Status : FAILED
Error: Java heap space
23/03/02 00:16:22 INFO mapreduce.Job:  map 0% reduce 0%
23/03/02 00:16:27 INFO mapreduce.Job:  map 100% reduce 100%
23/03/02 00:16:27 INFO mapreduce.Job: Job job_1677686996211_0022 failed with state FAILED due to: Task failed task_1677686996211_0022_m_000000
Job failed as tasks failed. failedMaps:1 failedReduces:0

23/03/02 00:16:27 INFO mapreduce.Job: Counters: 16
        Job Counters
                Failed map tasks=4
                Killed reduce tasks=1
                Launched map tasks=4
                Other local map tasks=3
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=35038
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=35038
                Total time spent by all reduce tasks (ms)=0
                Total vcore-milliseconds taken by all map tasks=35038
                Total vcore-milliseconds taken by all reduce tasks=0
                Total megabyte-milliseconds taken by all map tasks=35878912
                Total megabyte-milliseconds taken by all reduce tasks=0
        Map-Reduce Framework
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
root@hadoop-master:~# |
```

Unfortunately, as you can see, I got the Java Heap Space error, I tried to increase the memory of the JVM but it didn't work.

Let's try a seq file locally.



```
~/l/b/labs/homework on main !9 ?2 ❯  /usr/bin/env /home/adam/.sdkman/candidates/java/8.0.282.j9-adpt/bin/java -cp /tmp/cp_
f2nu587fs1zcd0enk0zzoz1bn.jar tn.insat.tp1.WordCount src/main/resources/input/file.seq src/main/resources/output
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
```



```
1 Canada-Thornhill
2 United States-San Marcos
3 United States-Wilmington
4 United States-Palo Alto
5 United States-Clifton
6 United States-Ashburn
7 United States-North Bergen
8 United States-Hampton
9 United States-Meredosia
10 United States-Seattle
11 United States-Hampton
12 United States-Mountain View  High anonymity  2886 kB/s
13 United States-Houston
14 United States-Ashburn
15 United States-Ashburn
```

6 Conclusion

In this lab, I learned how to install Hadoop on a cluster of 3 nodes, and how to run a simple MapReduce job using Hadoop. I also learned how to use the Hadoop Web UI to monitor the job.

Lab By: Mrs. Rabaa Youssef.