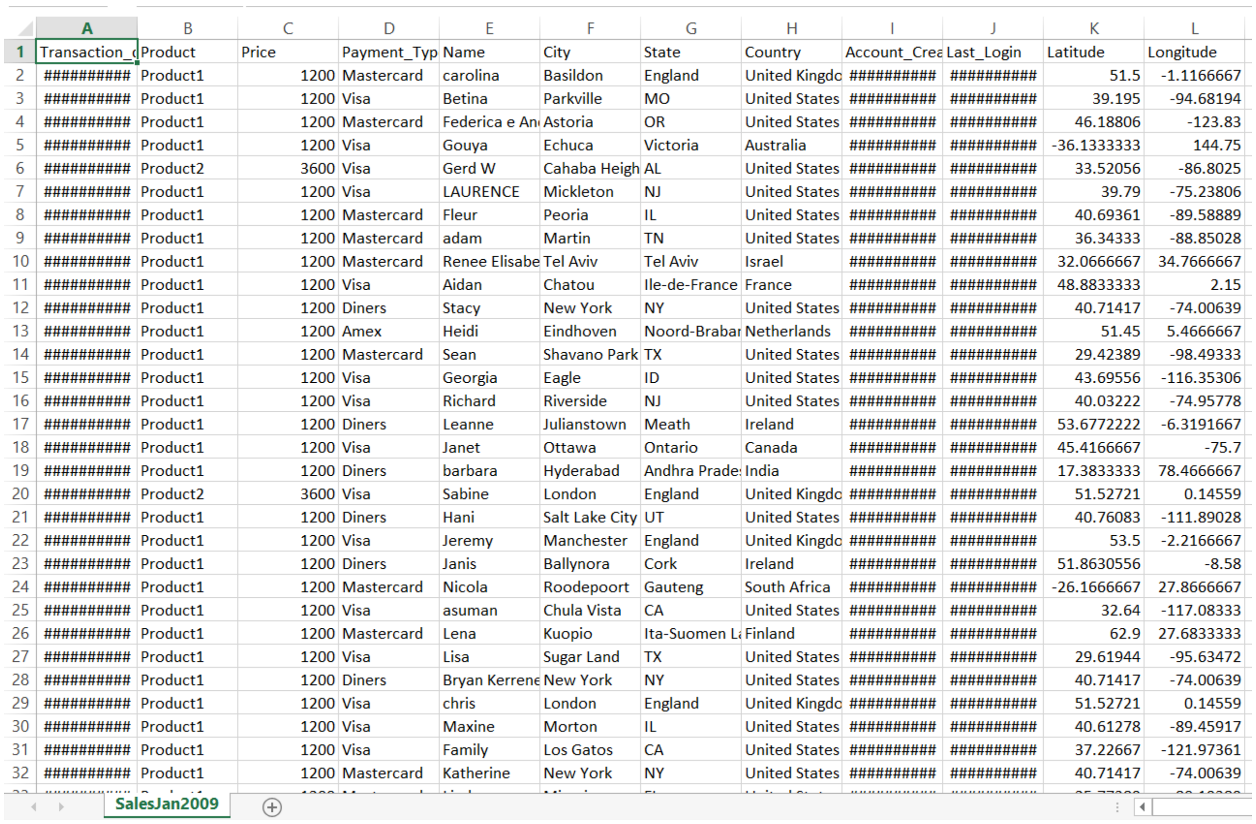
Case study: Analysis of the sale of certain products in the world

In this project, I will use Hadoop with MapReduce. The input data that we are going to use is a sale products dataset calls SalesJa2009.csv that we can find on kaggle website by clicking on this link: https://www.kaggle.com/jensroderus/salesjan2009csv. It contains Sales related information like product name, price, payment mode, city, country of client etc. The goal is to **find out the Number of Products Sold in Each Country.**



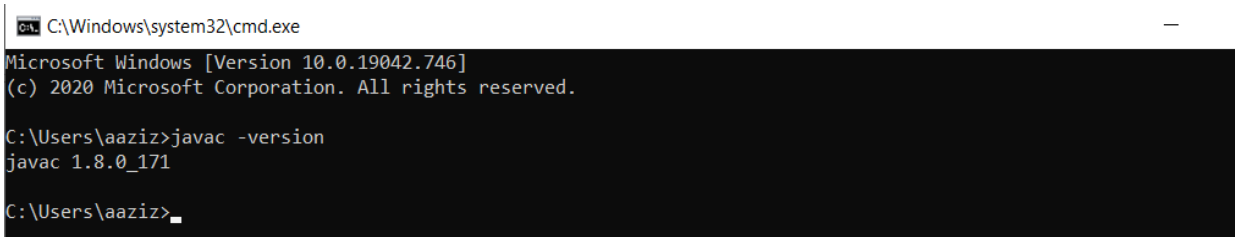
	A	B	C	D	E	F	G	H	I	J	K	L
1	Transaction_d	Product	Price	Payment_Typ	Name	City	State	Country	Account_Crea	Last_Login	Latitude	Longitude
2	##########	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdo	##########	##########	51.5	-1.1166667
3	##########	Product1	1200	Visa	Betina	Parkville	MO	United States	##########	##########	39.195	-94.68194
4	##########	Product1	1200	Mastercard	Federica e An	Astoria	OR	United States	##########	##########	46.18806	-123.83
5	##########	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	##########	##########	-36.1333333	144.75
6	##########	Product2	3600	Visa	Gerd W	Cahaba Heigh	AL	United States	##########	##########	33.52056	-86.8025
7	##########	Product1	1200	Visa	LAURENCE	Mickleton	NJ	United States	##########	##########	39.79	-75.23806
8	##########	Product1	1200	Mastercard	Fleur	Peoria	IL	United States	##########	##########	40.69361	-89.58889
9	##########	Product1	1200	Mastercard	adam	Martin	TN	United States	##########	##########	36.34333	-88.85028
10	##########	Product1	1200	Mastercard	Renee Elisabe	Tel Aviv	Tel Aviv	Israel	##########	##########	32.0666667	34.7666667
11	##########	Product1	1200	Visa	Aidan	Chatou	Ile-de-France	France	##########	##########	48.8833333	2.15
12	##########	Product1	1200	Diners	Stacy	New York	NY	United States	##########	##########	40.71417	-74.00639
13	##########	Product1	1200	Amex	Heidi	Eindhoven	Noord-Brabar	Netherlands	##########	##########	51.45	5.4666667
14	##########	Product1	1200	Mastercard	Sean	Shavano Park	TX	United States	##########	##########	29.42389	-98.49333
15	##########	Product1	1200	Visa	Georgia	Eagle	ID	United States	##########	##########	43.69556	-116.35306
16	##########	Product1	1200	Visa	Richard	Riverside	NJ	United States	##########	##########	40.03222	-74.95778
17	##########	Product1	1200	Diners	Leanne	Julianstown	Meath	Ireland	##########	##########	53.6772222	-6.3191667
18	##########	Product1	1200	Visa	Janet	Ottawa	Ontario	Canada	##########	##########	45.4166667	-75.7
19	##########	Product1	1200	Diners	barbara	Hyderabad	Andhra Prade:	India	##########	##########	17.3833333	78.4666667
20	##########	Product2	3600	Visa	Sabine	London	England	United Kingdo	##########	##########	51.52721	0.14559
21	##########	Product1	1200	Diners	Hani	Salt Lake City	UT	United States	##########	##########	40.76083	-111.89028
22	##########	Product1	1200	Visa	Jeremy	Manchester	England	United Kingdo	##########	##########	53.5	-2.2166667
23	##########	Product1	1200	Diners	Janis	Ballynora	Cork	Ireland	##########	##########	51.8630556	-8.58
24	##########	Product1	1200	Mastercard	Nicola	Roodepoort	Gauteng	South Africa	##########	##########	-26.1666667	27.8666667
25	##########	Product1	1200	Visa	asuman	Chula Vista	CA	United States	##########	##########	32.64	-117.08333
26	##########	Product1	1200	Mastercard	Lena	Kuopio	Ita-Suomen Lä	Finland	##########	##########	62.9	27.6833333
27	##########	Product1	1200	Visa	Lisa	Sugar Land	TX	United States	##########	##########	29.61944	-95.63472
28	##########	Product1	1200	Diners	Bryan Kerrene	New York	NY	United States	##########	##########	40.71417	-74.00639
29	##########	Product1	1200	Visa	chris	London	England	United Kingdo	##########	##########	51.52721	0.14559
30	##########	Product1	1200	Visa	Maxine	Morton	IL	United States	##########	##########	40.61278	-89.45917
31	##########	Product1	1200	Visa	Family	Los Gatos	CA	United States	##########	##########	37.22667	-121.97361
32	##########	Product1	1200	Mastercard	Katherine	New York	NY	United States	##########	##########	40.71417	-74.00639

SalesJan2009

First, we will start by installing the hadoop framework in our Windows 10 machine, then configure it before starting to write our java mapreduce code for our case study.

I- Installation of Hadoop

We assume that java version 1.8 is already installed on our computer. In our case, version 1.8.0_171 is installed in our machine.



C:\Windows\system32\cmd.exe

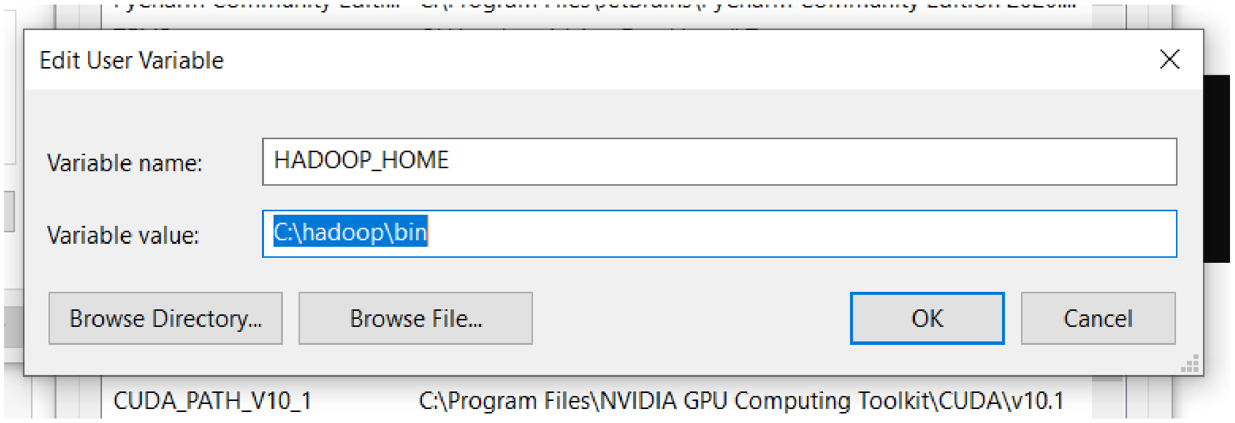
```
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\aaziz>javac -version
javac 1.8.0_171

C:\Users\aaziz>_
```

To install Hadoop in our windows machine, we need to follow the following steps:

Step 1: Download Hadoop binary package from apache website (https://hadoop.apache.org/releases.html)

In our case we have downloaded hadoop version 2.7.6

Step 2: unpack the package, copy the folder in our C directory and configure the path in our environment variable.

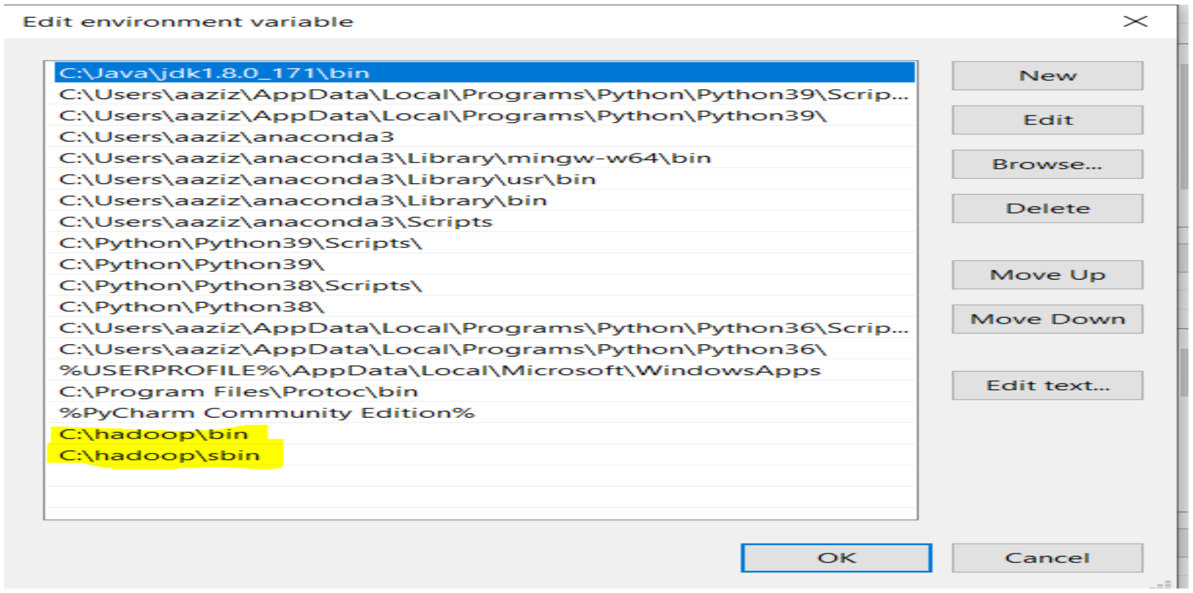


Edit User Variable

Variable name: HADOOP_HOME

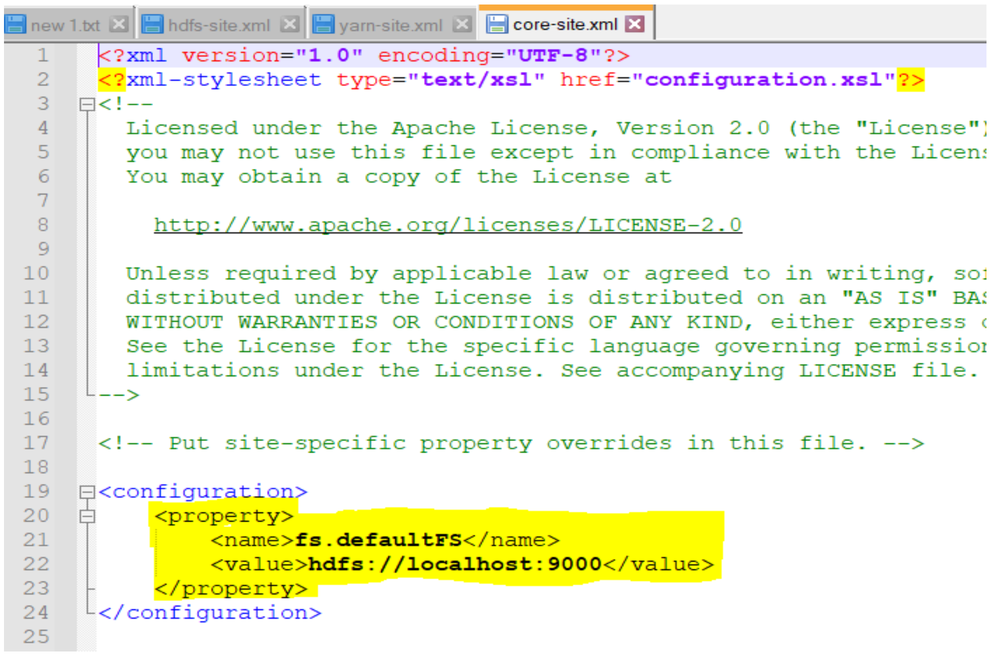
Variable value: C:\hadoop\bin

CUDA_PATH_V10_1   C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1

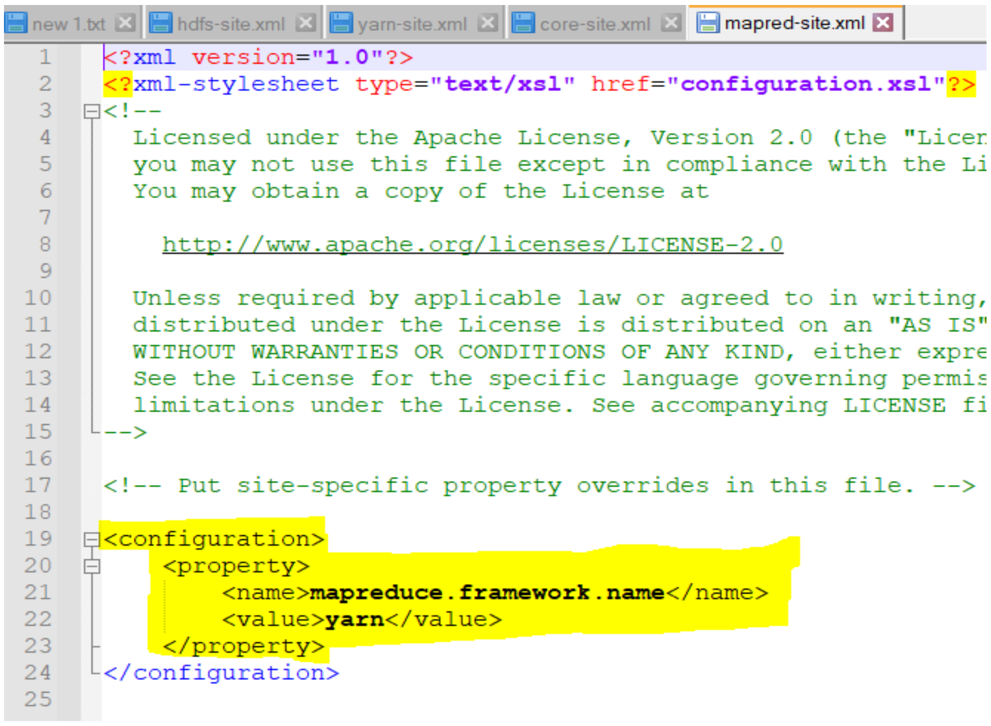


```
C:\Java\jdk1.8.0_171\bin
C:\Users\aaziz\AppData\Local\Programs\Python\Python39\Scrip...
C:\Users\aaziz\AppData\Local\Programs\Python\Python39\
C:\Users\aaziz\anaconda3
C:\Users\aaziz\anaconda3\Library\mingw-w64\bin
C:\Users\aaziz\anaconda3\Library\usr\bin
C:\Users\aaziz\anaconda3\Library\bin
C:\Users\aaziz\anaconda3\Scripts
C:\Python\Python39\Scripts\
C:\Python\Python39\
C:\Python\Python38\Scripts\
C:\Python\Python38\
C:\Users\aaziz\AppData\Local\Programs\Python\Python36\Scrip...
C:\Users\aaziz\AppData\Local\Programs\Python\Python36\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\Protoc\bin
%PyCharm Community Edition%
C:\hadoop\bin
C:\hadoop\sbin
```

Step 3: modification of core-site.xml, mapred-site.xml, hdfs-site.xml and yarn-site.xml which are in C: //hadoop/etc/hadoop.

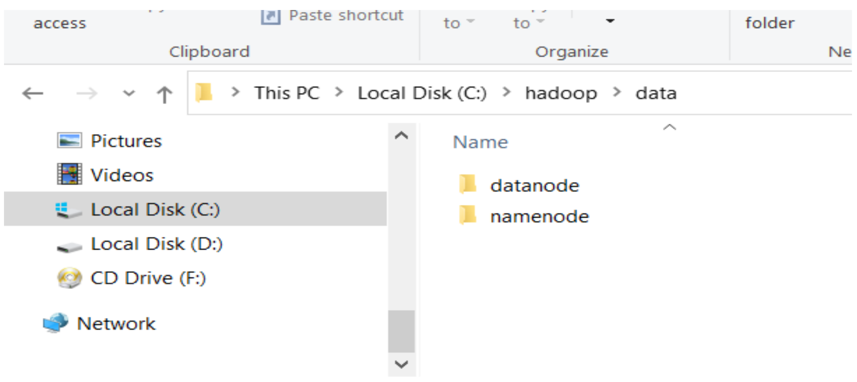


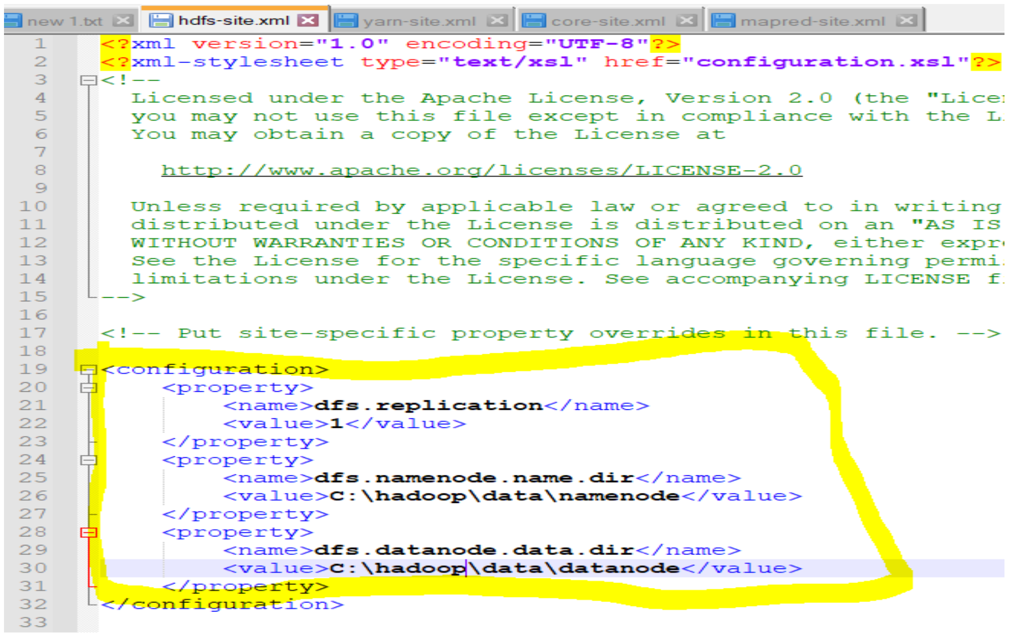
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License")
  you may not use this file except in compliance with the Licen
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, so
  distributed under the License is distributed on an "AS IS" BA
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express 
  See the License for the specific language governing permissio
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>

```



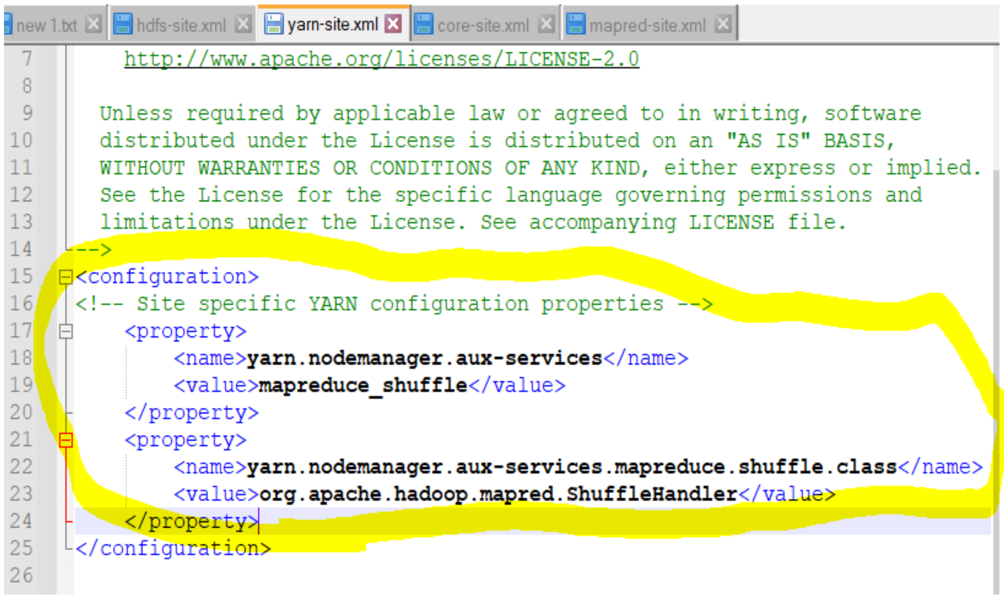
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "Licen
  you may not use this file except in compliance with the Li
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
  distributed under the License is distributed on an "AS IS"
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either expre
  See the License for the specific language governing permis
  limitations under the License. See accompanying LICENSE fi
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

Before modifying hdfs-site.xml and yarn-site.xml, we must create a folder in C: //hadoop calls data and create two more folders (datanode and namenode).




new 1.txt | hdfs-site.xml | yarn-site.xml | core-site.xml | mapred-site.xml

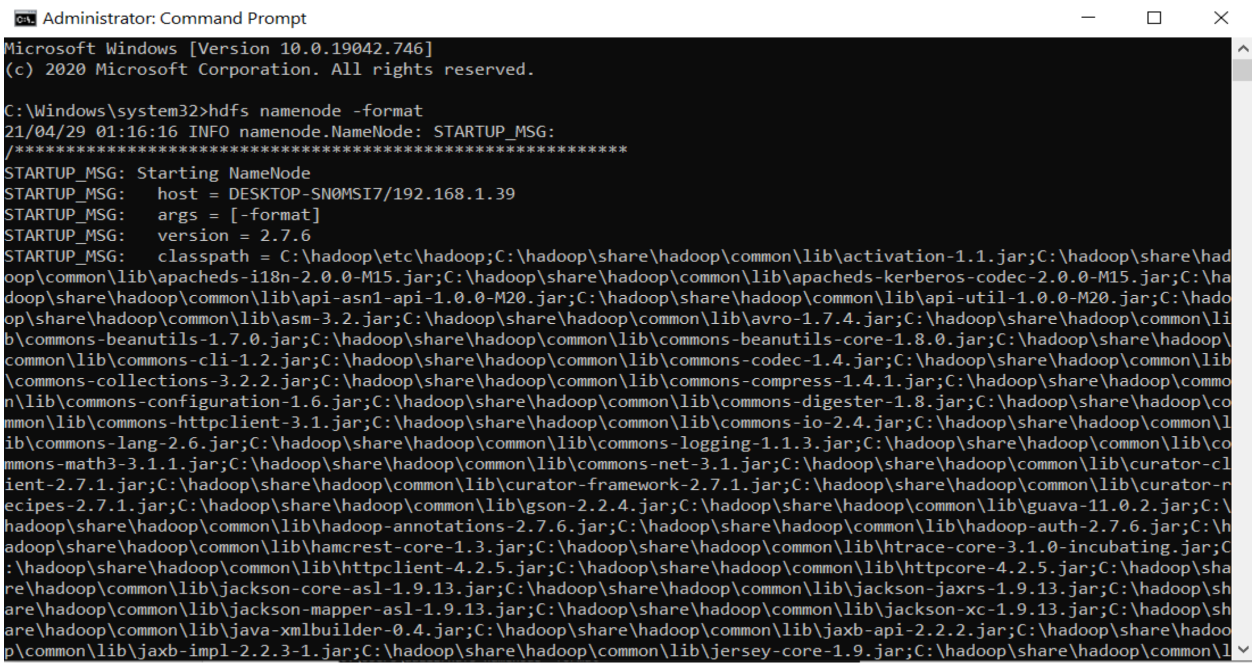
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "Lice
  you may not use this file except in compliance with the L
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing
  distributed under the License is distributed on an "AS IS
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either expr
  See the License for the specific language governing permi
  limitations under the License. See accompanying LICENSE f
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>C:\hadoop\data\namenode</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>C:\hadoop\data\datanode</value>
    </property>
</configuration>
```


new 1.txt | hdfs-site.xml | yarn-site.xml | core-site.xml | mapred-site.xml

```
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
</configuration>
```

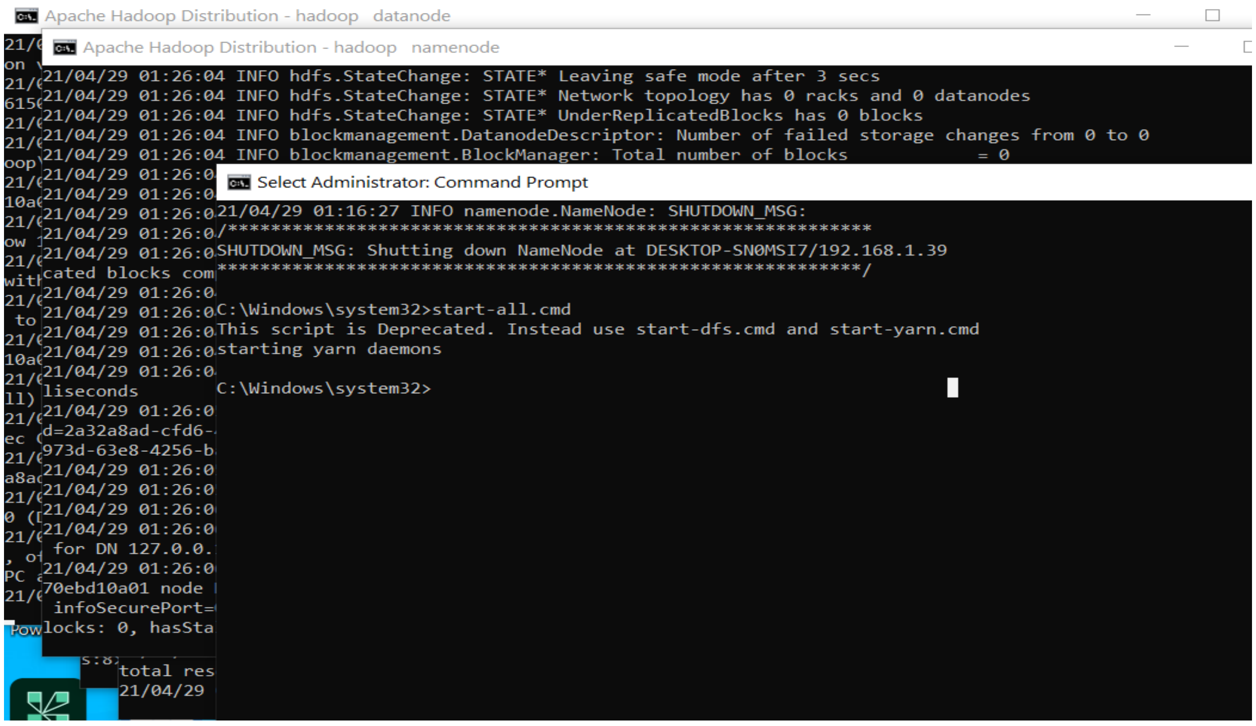
Step 4: Format the namenode and start all daemons

To format the namenode, we will use the following command: **hdfs namenode -format**



```
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Windows\system32>hdfs namenode -format
21/04/29 01:16:16 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-SN0MSI7/192.168.1.39
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.6
STARTUP_MSG:   classpath = C:\hadoop\etc\hadoop;C:\hadoop\share\hadoop\common\lib\activation-1.1.jar;C:\hadoop\share\had
oop\common\lib\apacheds-i18n-2.0.0-M15.jar;C:\hadoop\share\hadoop\common\lib\apacheds-kerberos-codec-2.0.0-M15.jar;C:\ha
doop\share\hadoop\common\lib\api-asn1-api-1.0.0-M20.jar;C:\hadoop\share\hadoop\common\lib\api-util-1.0.0-M20.jar;C:\hado
op\share\hadoop\common\lib\asm-3.2.jar;C:\hadoop\share\hadoop\common\lib\avro-1.7.4.jar;C:\hadoop\share\hadoop\common\li
b\commons-beanutils-1.7.0.jar;C:\hadoop\share\hadoop\common\lib\commons-beanutils-core-1.8.0.jar;C:\hadoop\share\hadoop\
common\lib\commons-cli-1.2.jar;C:\hadoop\share\hadoop\common\lib\commons-codec-1.4.jar;C:\hadoop\share\hadoop\common\lib
\commons-collections-3.2.2.jar;C:\hadoop\share\hadoop\common\lib\commons-compress-1.4.1.jar;C:\hadoop\share\hadoop\commo
n\lib\commons-configuration-1.6.jar;C:\hadoop\share\hadoop\common\lib\commons-digester-1.8.jar;C:\hadoop\share\hadoop\co
mmon\lib\commons-httpclient-3.1.jar;C:\hadoop\share\hadoop\common\lib\commons-io-2.4.jar;C:\hadoop\share\hadoop\common\l
ib\commons-lang-2.6.jar;C:\hadoop\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop\share\hadoop\common\lib\co
mmons-math3-3.1.1.jar;C:\hadoop\share\hadoop\common\lib\commons-net-3.1.jar;C:\hadoop\share\hadoop\common\lib\curator-cl
ient-2.7.1.jar;C:\hadoop\share\hadoop\common\lib\curator-framework-2.7.1.jar;C:\hadoop\share\hadoop\common\lib\curator-r
ecipes-2.7.1.jar;C:\hadoop\share\hadoop\common\lib\gson-2.2.4.jar;C:\hadoop\share\hadoop\common\lib\guava-11.0.2.jar;C:\
hadoop\share\hadoop\common\lib\hadoop-annotations-2.7.6.jar;C:\hadoop\share\hadoop\common\lib\hadoop-auth-2.7.6.jar;C:\h
adoop\share\hadoop\common\lib\hamcrest-core-1.3.jar;C:\hadoop\share\hadoop\common\lib\htrace-core-3.1.0-incubating.jar;C
:\hadoop\share\hadoop\common\lib\httpclient-4.2.5.jar;C:\hadoop\share\hadoop\common\lib\httpcore-4.2.5.jar;C:\hadoop\sha
re\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop\sh
are\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop\share\hadoop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop\sh
are\hadoop\common\lib\java-xmlbuilder-0.4.jar;C:\hadoop\share\hadoop\common\lib\jaxb-api-2.2.2.jar;C:\hadoop\share\hadoo
p\common\lib\jaxb-impl-2.2.3-1.jar;C:\hadoop\share\hadoop\common\lib\jersey-core-1.9.jar;C:\hadoop\share\hadoop\common\l
```

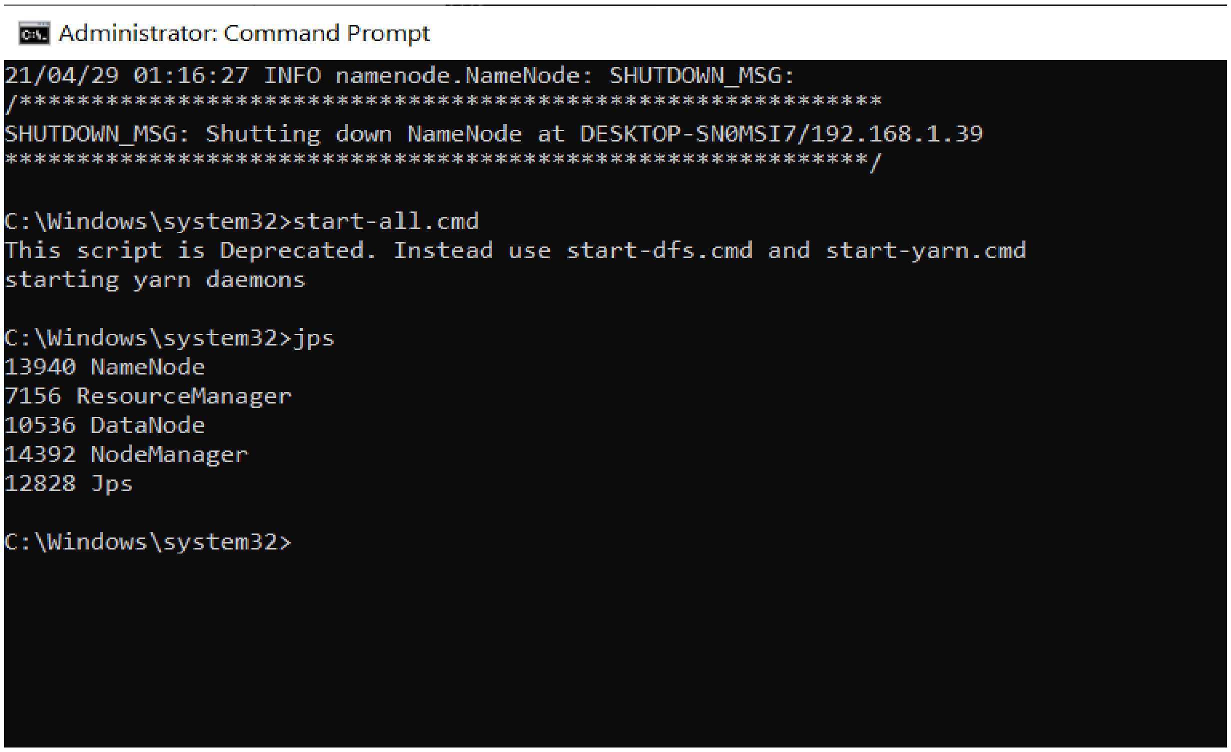
Our namenode has started successfully. We are now going to start our daemons which are: namenode, datanode, resourcemanager and nodemanager. We will use the following command: **start-all.cmd**


Apache Hadoop Distribution - hadoop datanode
Apache Hadoop Distribution - hadoop namenode
Select Administrator: Command Prompt

```
21/04/29 01:26:04 INFO hdfs.StateChange: STATE* Leaving safe mode after 3 secs
21/04/29 01:26:04 INFO hdfs.StateChange: STATE* Network topology has 0 racks and 0 datanodes
21/04/29 01:26:04 INFO hdfs.StateChange: STATE* UnderReplicatedBlocks has 0 blocks
21/04/29 01:26:04 INFO blockmanagement.DatanodeDescriptor: Number of failed storage changes from 0 to 0
21/04/29 01:26:04 INFO blockmanagement.BlockManager: Total number of blocks            = 0
```

```
21/04/29 01:16:27 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-SN0MSI7/192.168.1.39
************************************************************/

C:\Windows\system32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\system32>
```

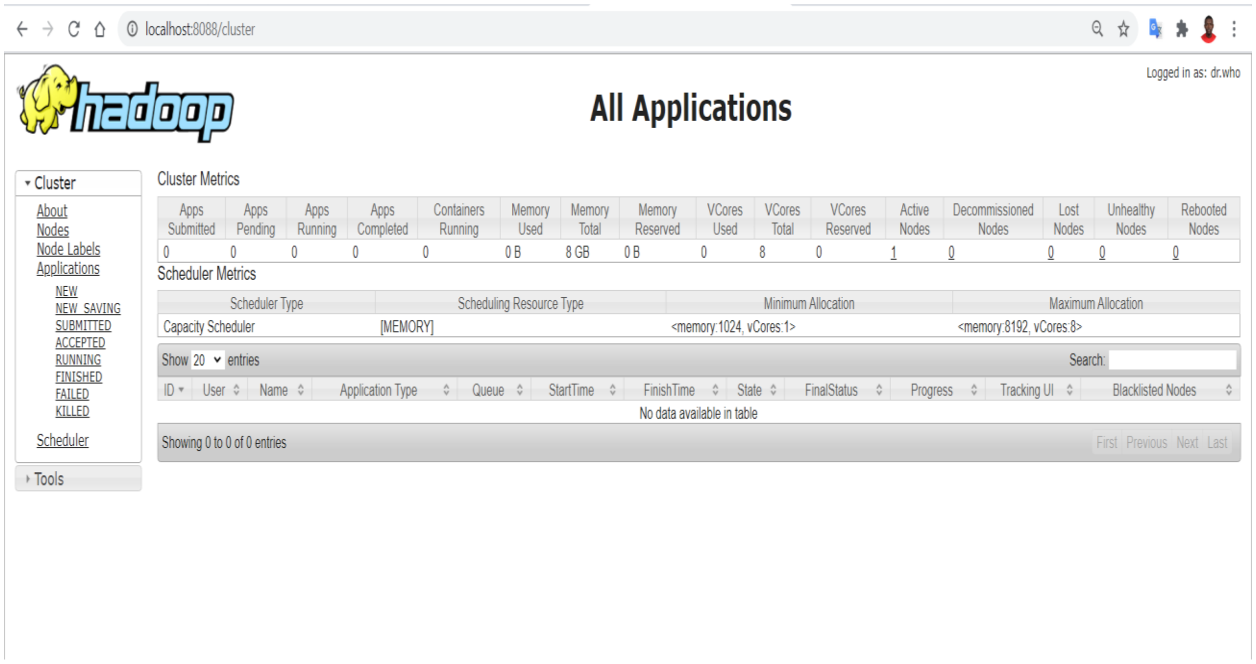
The **jps** command allows us to check the demons that are started on our machine.



Administrator: Command Prompt

```
21/04/29 01:16:27 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-SN0MSI7/192.168.1.39
************************************************************/

C:\Windows\system32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\system32>jps
13940 NameNode
7156 ResourceManager
10536 DataNode
14392 NodeManager
12828 Jps

C:\Windows\system32>
```

We can see that all of our demons are cast. Now that we're done with that, let's see our cluster and our namenode in localhost. The address page for our is **localhost:8088** and for our namenode page is **localhost:50070**



localhost:8088/cluster

Logged in as: dr.who

All Applications

Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools

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	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

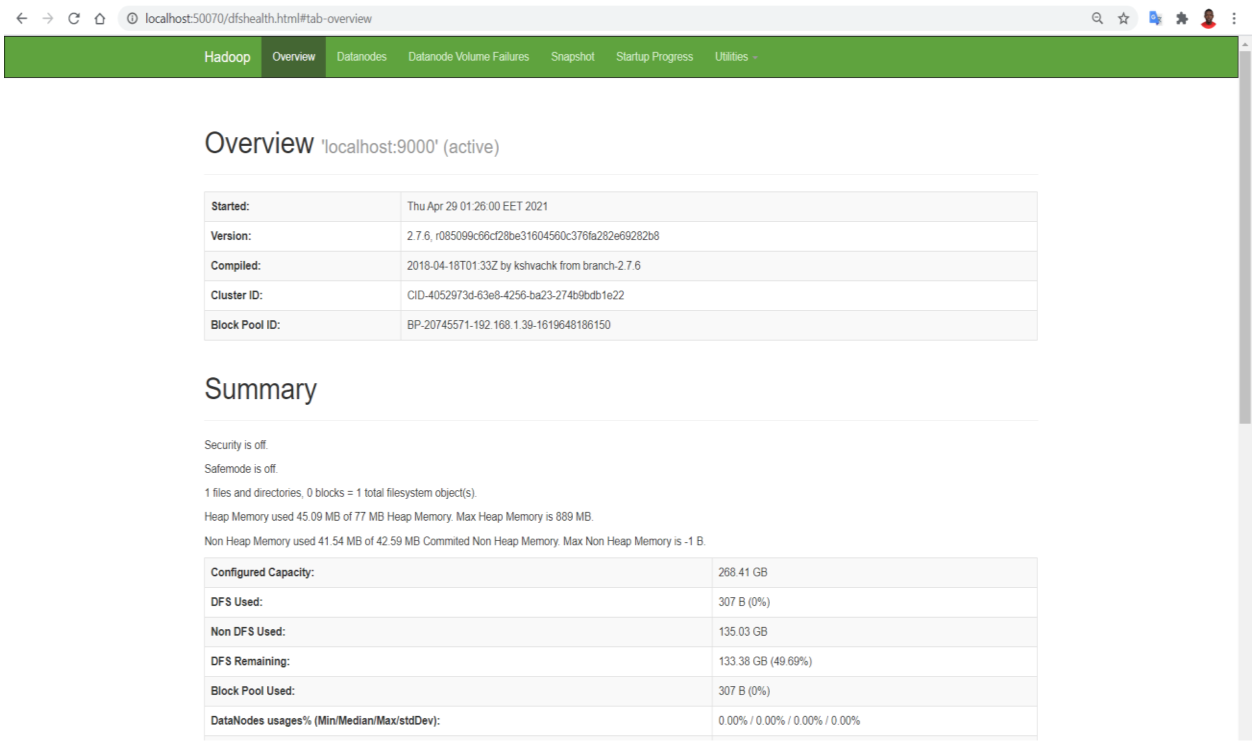
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
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	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries   First Previous Next Last



Overview 'localhost:9000' (active)

Started:	Thu Apr 29 01:26:00 EET 2021
Version:	2.7.6, r085099c66cf28be31604560c376fa282e69282b8
Compiled:	2018-04-18T01:33Z by kshvachk from branch-2.7.6
Cluster ID:	CID-4052973d-63e8-4256-ba23-274b9bdb1e22
Block Pool ID:	BP-20745571-192.168.1.39-1619648186150

Summary

Security is off.

Safemode is off.

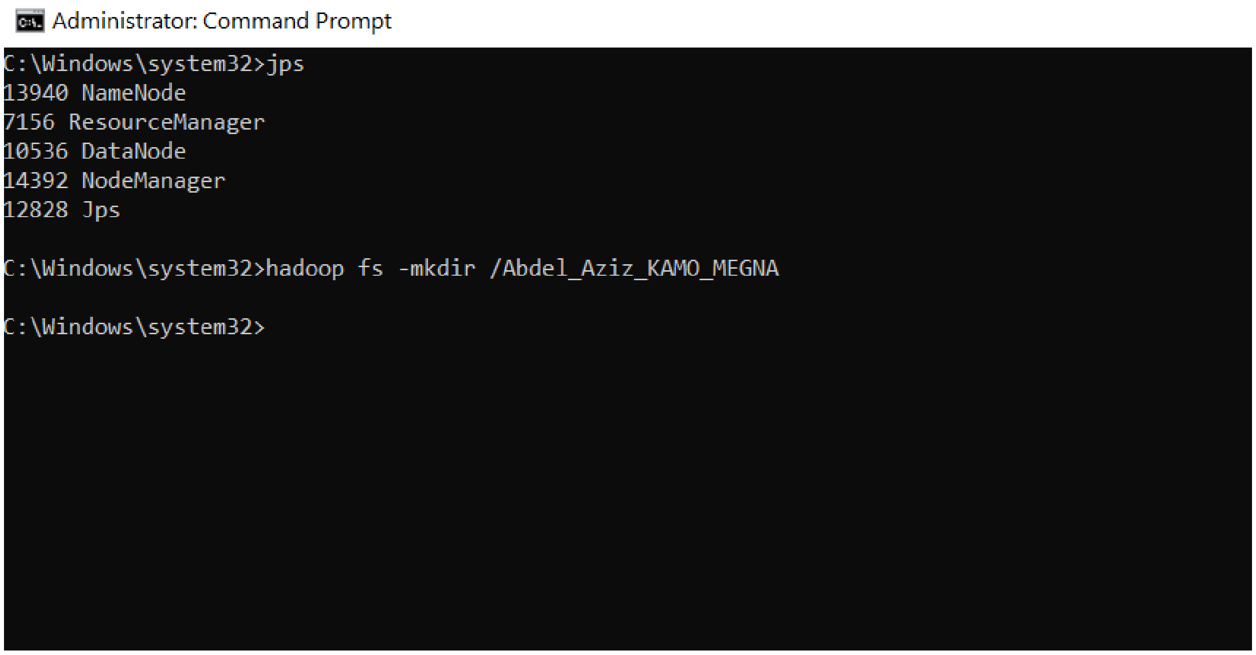
1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 45.09 MB of 77 MB Heap Memory. Max Heap Memory is 889 MB.

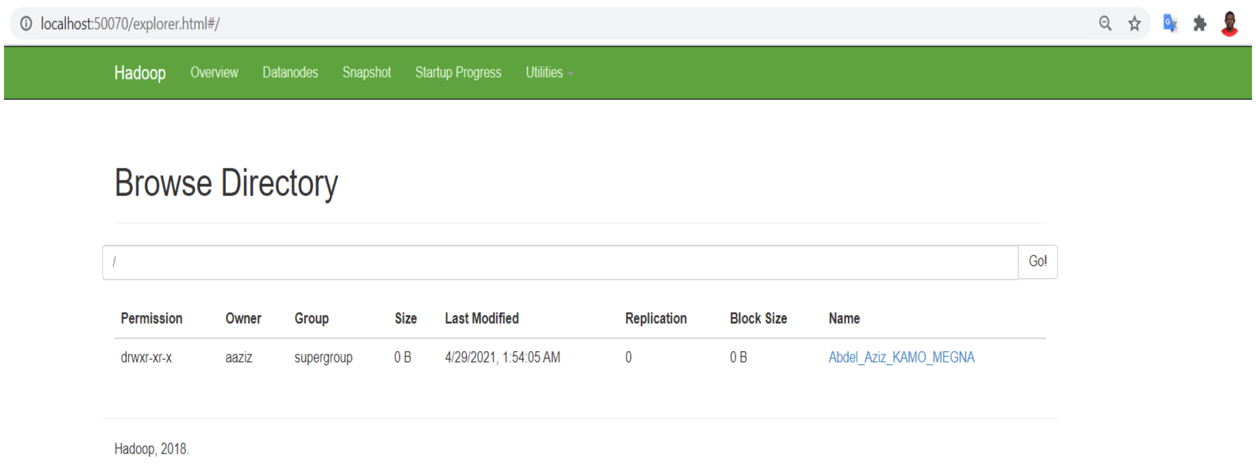
Non Heap Memory used 41.54 MB of 42.59 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	268.41 GB
DFS Used:	307 B (0%)
Non DFS Used:	135.03 GB
DFS Remaining:	133.38 GB (49.69%)
Block Pool Used:	307 B (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

Now we are going to set up a folder name Abdel_Aziz_KAMO_MEGNA (which is my name) in our hadoop space by using command line: **hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA**



```
C:\Windows\system32>jps
13940 NameNode
7156 ResourceManager
10536 DataNode
14392 NodeManager
12828 Jps

C:\Windows\system32>hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA

C:\Windows\system32>
```



Browse Directory

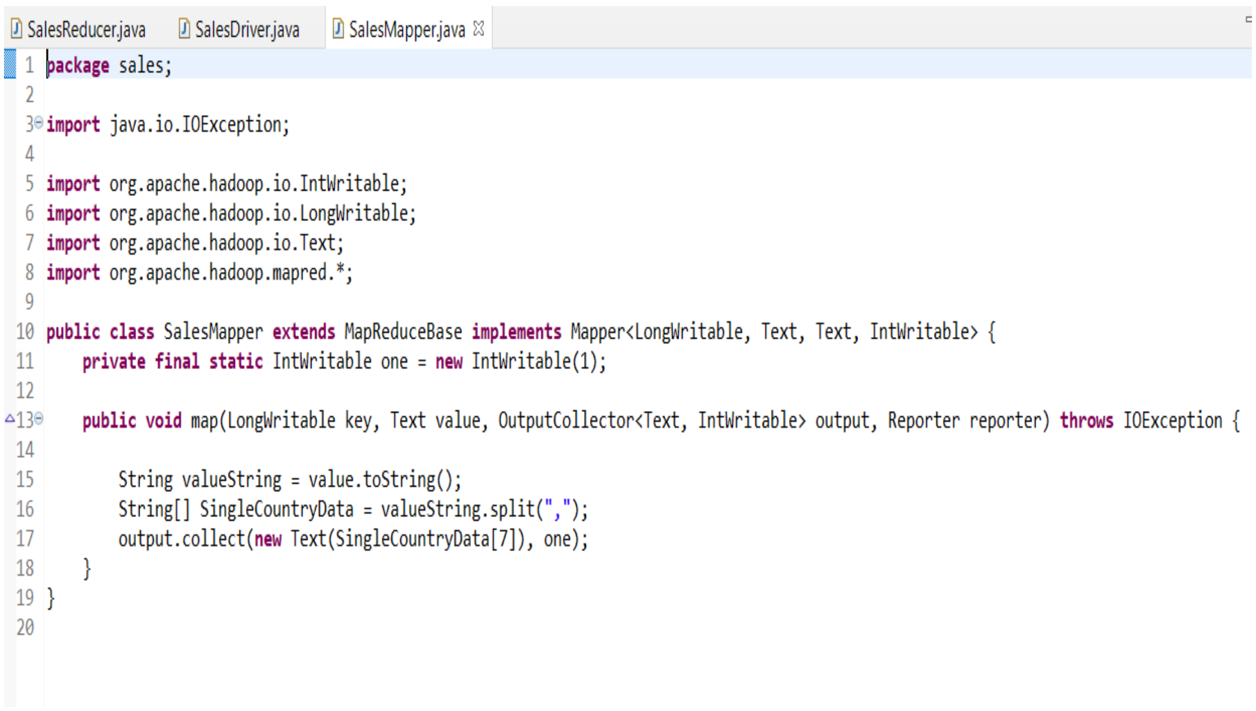
/

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	aaziz	supergroup	0 B	4/29/2021, 1:54:05 AM	0	0 B	Abdel_Aziz_KAMO_MEGNA

Hadoop, 2018.

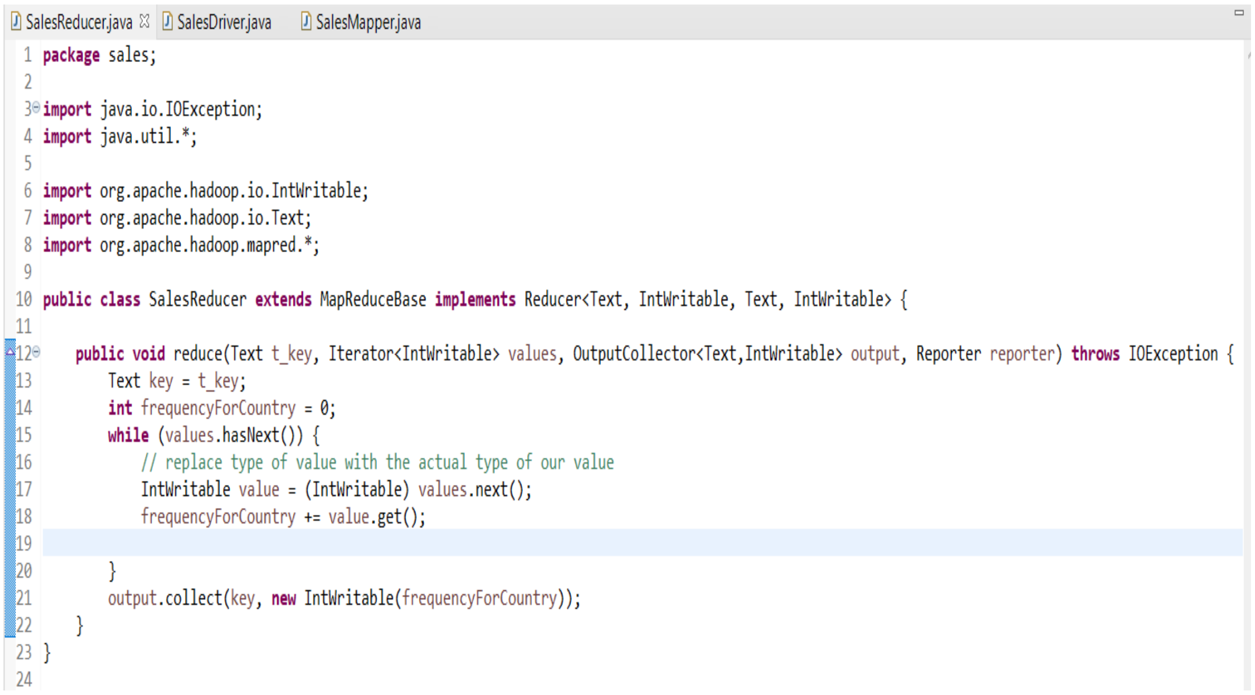
II- Map, Reduce and Driver code for our case study

1- SalesMapper.java



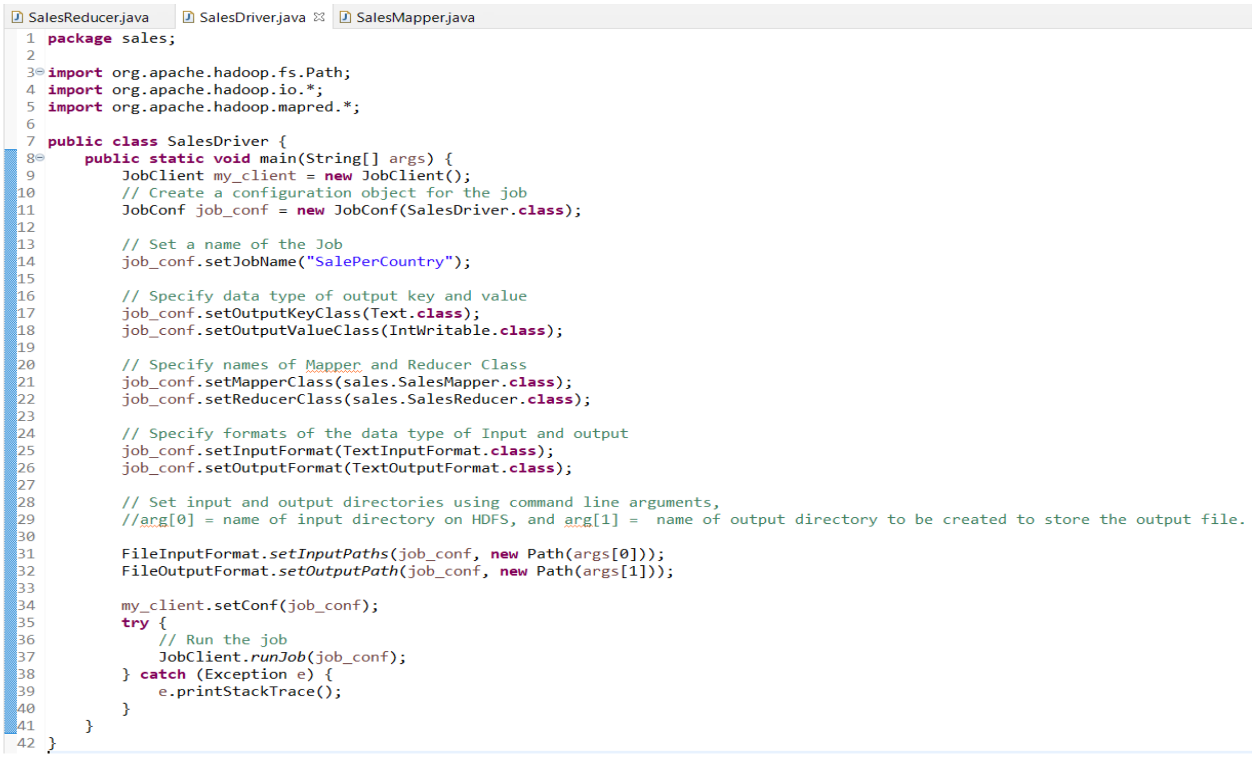
```java
package sales;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);

    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

        String valueString = value.toString();
        String[] SingleCountryData = valueString.split(",");
        output.collect(new Text(SingleCountryData[7]), one);
    }
}
```

2- SalesReduce.java



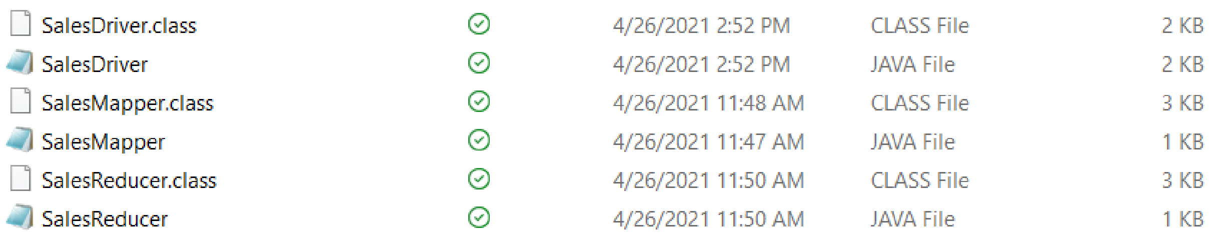
```java
package sales;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text t_key, Iterator<IntWritable> values, OutputCollector<Text,IntWritable> output, Reporter reporter) throws IOException {
        Text key = t_key;
        int frequencyForCountry = 0;
        while (values.hasNext()) {
            // replace type of value with the actual type of our value
            IntWritable value = (IntWritable) values.next();
            frequencyForCountry += value.get();

        }
        output.collect(key, new IntWritable(frequencyForCountry));
    }
}

```

3- SalesDriver.java



```java
package sales;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class SalesDriver {
    public static void main(String[] args) {
        JobClient my_client = new JobClient();
        // Create a configuration object for the job
        JobConf job_conf = new JobConf(SalesDriver.class);

        // Set a name of the Job
        job_conf.setJobName("SalePerCountry");

        // Specify data type of output key and value
        job_conf.setOutputKeyClass(Text.class);
        job_conf.setOutputValueClass(IntWritable.class);

        // Specify names of Mapper and Reducer Class
        job_conf.setMapperClass(sales.SalesMapper.class);
        job_conf.setReducerClass(sales.SalesReducer.class);

        // Specify formats of the data type of Input and output
        job_conf.setInputFormat(TextInputFormat.class);
        job_conf.setOutputFormat(TextOutputFormat.class);

        // Set input and output directories using command line arguments,
        //arg[0] = name of input directory on HDFS, and arg[1] =  name of output directory to be created to store the output file.

        FileInputFormat.setInputPaths(job_conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(job_conf, new Path(args[1]));

        my_client.setConf(job_conf);
        try {
            // Run the job
            JobClient.runJob(job_conf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

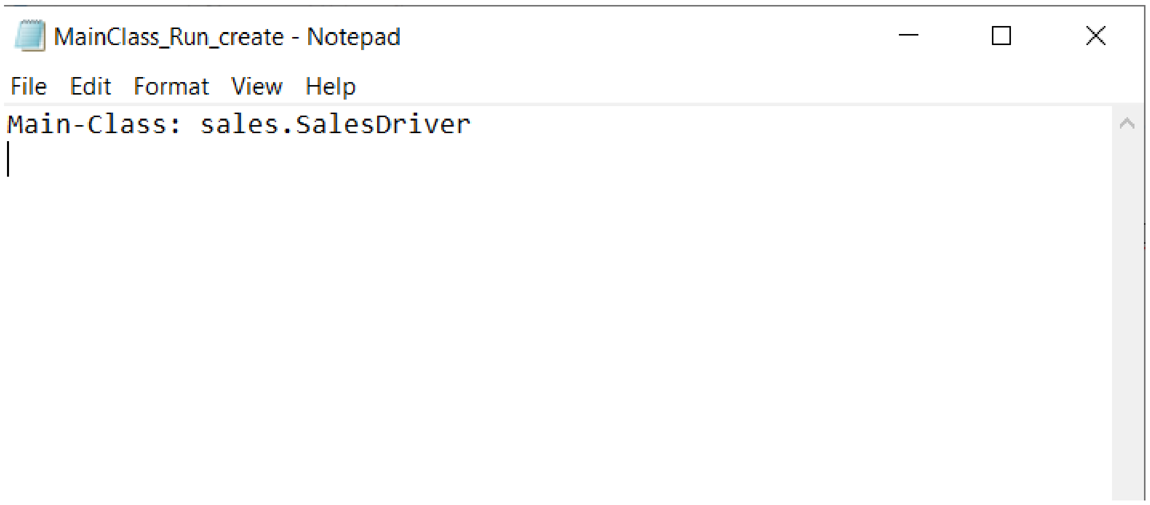
Once we have compile of our java files, the files: **SalesMapper.class, SalesReduce.class and SaleDriver.class** are automatically created.



SalesDriver.class	✓	4/26/2021 2:52 PM	CLASS File	2 KB
SalesDriver	✓	4/26/2021 2:52 PM	JAVA File	2 KB
SalesMapper.class	✓	4/26/2021 11:48 AM	CLASS File	3 KB
SalesMapper	✓	4/26/2021 11:47 AM	JAVA File	1 KB
SalesReducer.class	✓	4/26/2021 11:50 AM	CLASS File	3 KB
SalesReducer	✓	4/26/2021 11:50 AM	JAVA File	1 KB

4- Specification of our main class

We now need to specify our main class. This will be done by creating new text file (MainClass_run_create) and we will specify it inside.



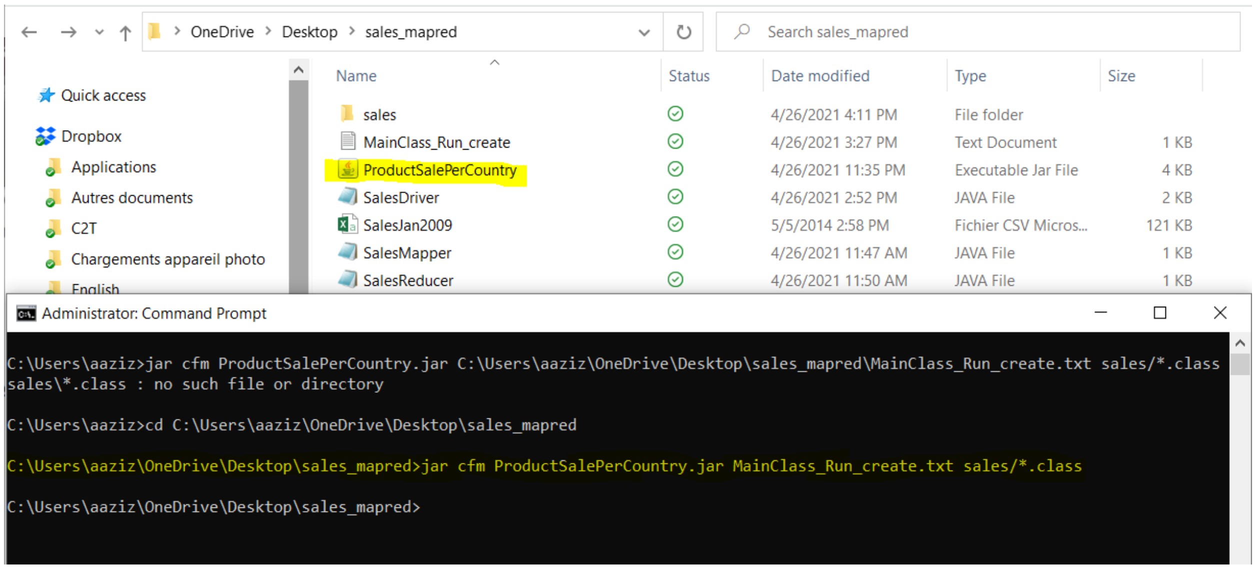
```
Main-Class: sales.SalesDriver
```

Sales.SaleDriver is the name of our main class

Note: we have to hit enter key at end of this line.

5- Create jar file for hadoop execution

This part consists of the creation of the jar file which will be executed by hadoop, this file contains the different classes of our application in which our main class is specified.

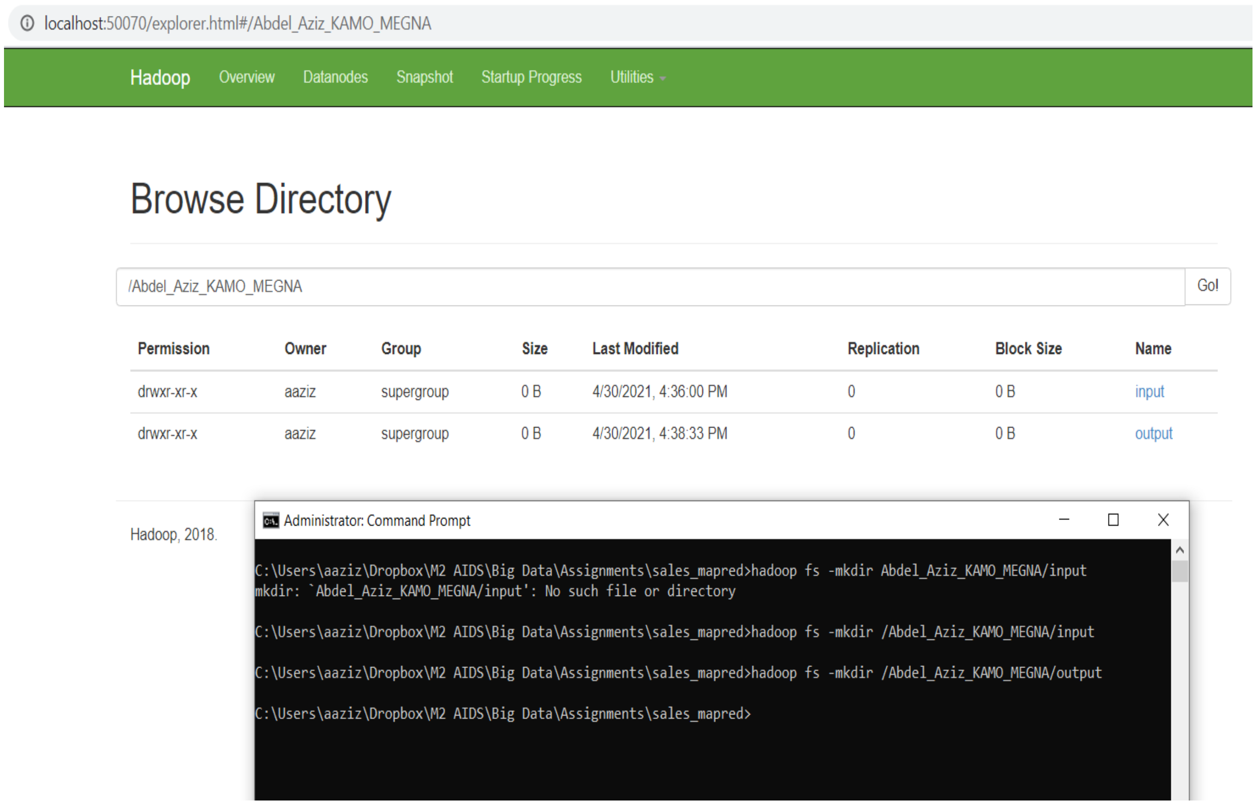

OneDrive > Desktop > sales_mapred
Search sales_mapred
Quick access
Dropbox
Applications
Autres documents
C2T
Chargements appareil photo
English
Administrator: Command Prompt

Name	Status	Date modified	Type	Size
sales	✓	4/26/2021 4:11 PM	File folder	
MainClass_Run_create	✓	4/26/2021 3:27 PM	Text Document	1 KB
ProductSalePerCountry	✓	4/26/2021 11:35 PM	Executable Jar File	4 KB
SalesDriver	✓	4/26/2021 2:52 PM	JAVA File	2 KB
SalesJan2009	✓	5/5/2014 2:58 PM	Fichier CSV Micros...	121 KB
SalesMapper	✓	4/26/2021 11:47 AM	JAVA File	1 KB
SalesReducer	✓	4/26/2021 11:50 AM	JAVA File	1 KB

```
C:\Users\aaziz>jar cfm ProductSalePerCountry.jar C:\Users\aaziz\OneDrive\Desktop\sales_mapred\MainClass_Run_create.txt sales/*.class
sales\*.class : no such file or directory

C:\Users\aaziz>cd C:\Users\aaziz\OneDrive\Desktop\sales_mapred

C:\Users\aaziz\OneDrive\Desktop\sales_mapred>jar cfm ProductSalePerCountry.jar MainClass_Run_create.txt sales/*.class

C:\Users\aaziz\OneDrive\Desktop\sales_mapred>
```

6- **Creation of our input and output directories in hadoop space**

- hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA/input
- hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA/output

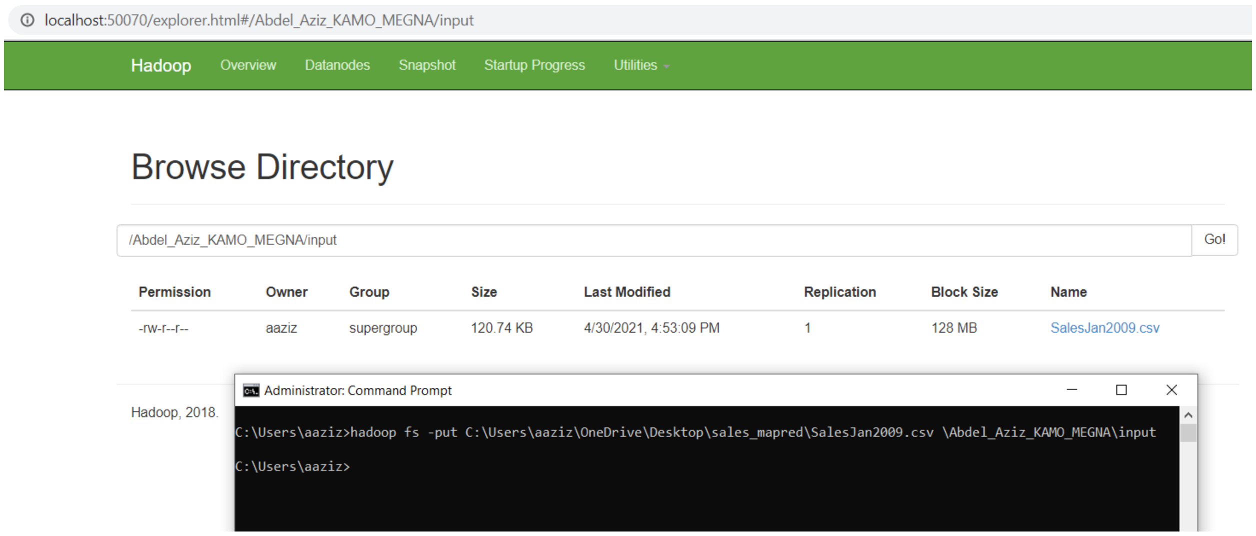


Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	aaziz	supergroup	0 B	4/30/2021, 4:36:00 PM	0	0 B	input
drwxr-xr-x	aaziz	supergroup	0 B	4/30/2021, 4:38:33 PM	0	0 B	output

```
C:\Users\aaziz\Dropbox\M2 AIDS\Big Data\Assignments\sales_mapred>hadoop fs -mkdir Abdel_Aziz_KAMO_MEGNA/input
mkdir: `Abdel_Aziz_KAMO_MEGNA/input': No such file or directory

C:\Users\aaziz\Dropbox\M2 AIDS\Big Data\Assignments\sales_mapred>hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA/input

C:\Users\aaziz\Dropbox\M2 AIDS\Big Data\Assignments\sales_mapred>hadoop fs -mkdir /Abdel_Aziz_KAMO_MEGNA/output

C:\Users\aaziz\Dropbox\M2 AIDS\Big Data\Assignments\sales_mapred>
```

7- Copy of our dataset file in our input directory

- hadoop fs -put C:\Users\aaziz\OneDrive\Desktop\sales_mapred\SalesJan2009.csv \Abdel_Aziz_KAMO_MEGNA\input

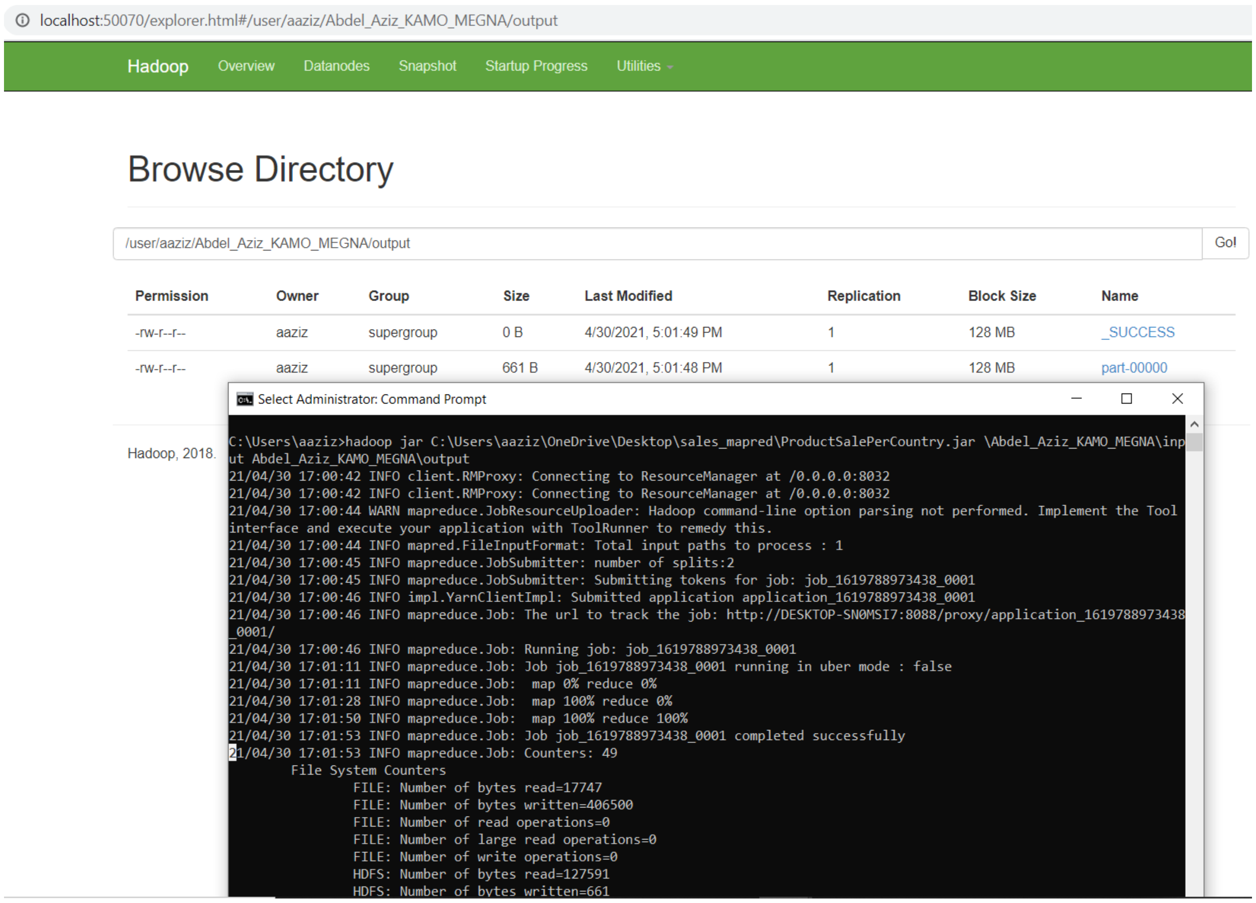

localhost:50070/explorer.html#/Abdel_Aziz_KAMO_MEGNA/input
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/Abdel_Aziz_KAMO_MEGNA/input
Go!
Hadoop, 2018.
Administrator: Command Prompt

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	aaziz	supergroup	120.74 KB	4/30/2021, 4:53:09 PM	1	128 MB	SalesJan2009.csv

```
C:\Users\aaziz>hadoop fs -put C:\Users\aaziz\OneDrive\Desktop\sales_mapred\SalesJan2009.csv \Abdel_Aziz_KAMO_MEGNA\input

C:\Users\aaziz>
```

8- Running our application

To run our application and put the result into our output folder, we have to execute this command: **hadoop jar ProductSalePerCountry.jar \Abdel_Aziz_KAMO_MEGNA\input Abdel_Aziz KAMO_MEGNA\output**



localhost:50070/explorer.html#/user/aaziz/Abdel_Aziz_KAMO_MEGNA/output

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

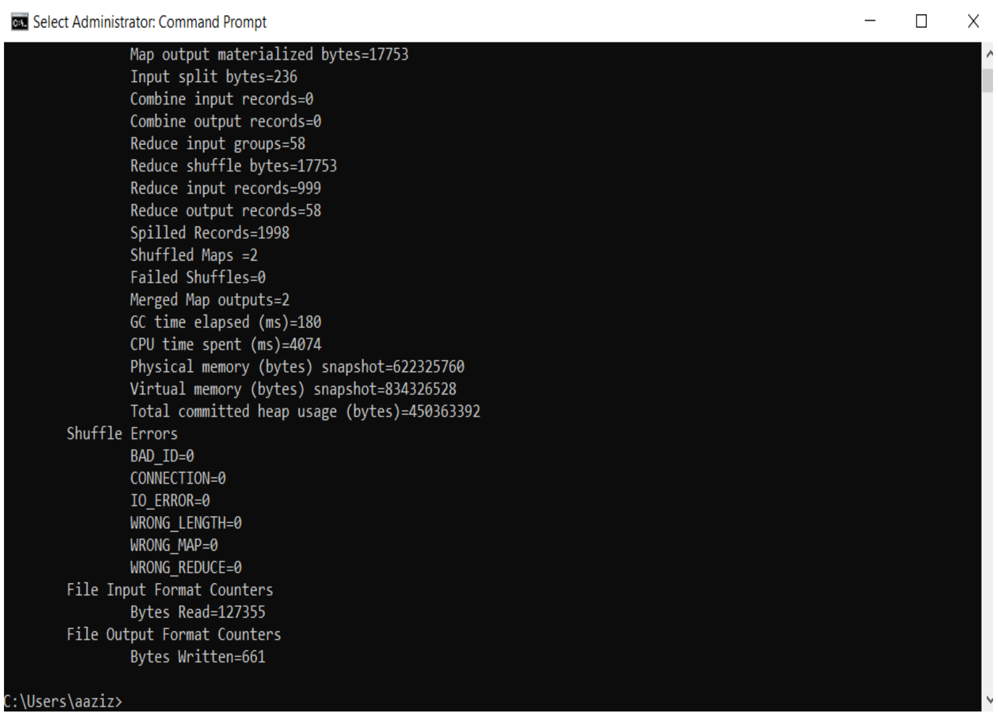
/user/aaziz/Abdel_Aziz_KAMO_MEGNA/output Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	aaziz	supergroup	0 B	4/30/2021, 5:01:49 PM	1	128 MB	_SUCCESS
-rw-r--r--	aaziz	supergroup	661 B	4/30/2021, 5:01:48 PM	1	128 MB	part-00000

Hadoop, 2018.

Select Administrator: Command Prompt

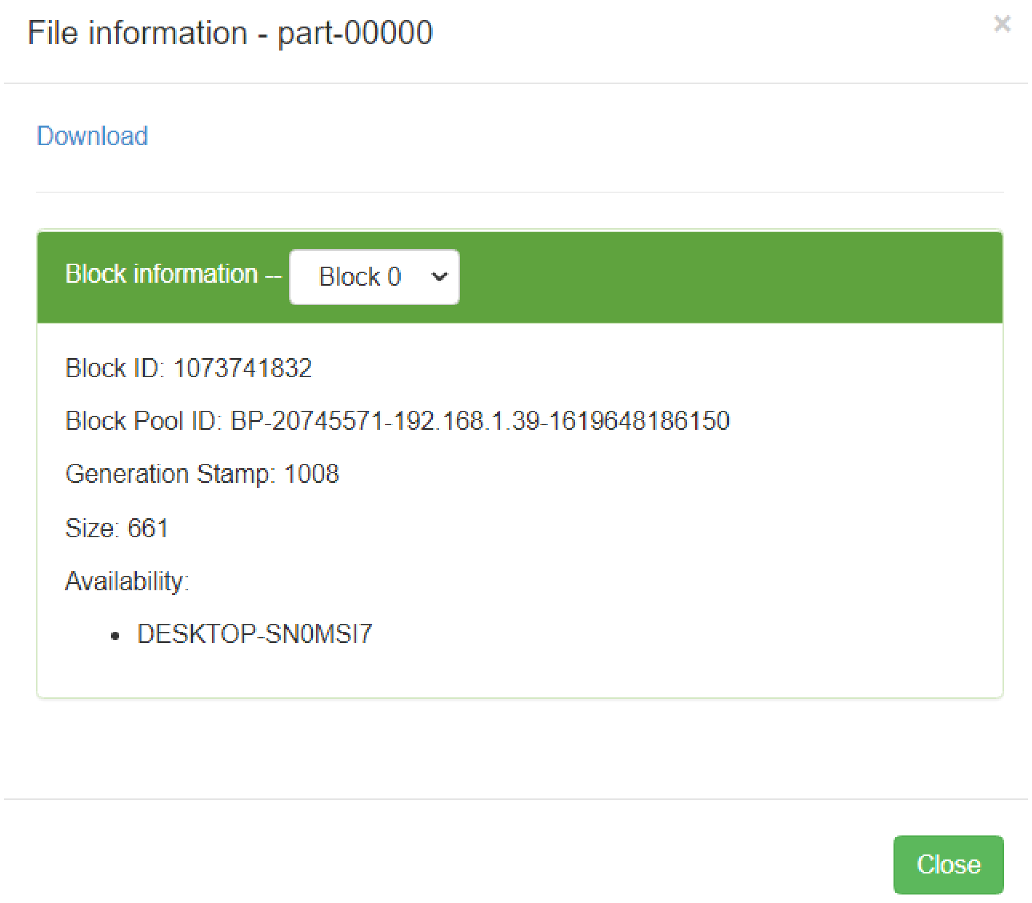
```
C:\Users\aaziz>hadoop jar C:\Users\aaziz\OneDrive\Desktop\sales_mapred\ProductSalePerCountry.jar \Abdel_Aziz_KAMO_MEGNA\inp
ut Abdel_Aziz_KAMO_MEGNA\output
21/04/30 17:00:42 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/04/30 17:00:42 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/04/30 17:00:44 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool
interface and execute your application with ToolRunner to remedy this.
21/04/30 17:00:44 INFO mapred.FileInputFormat: Total input paths to process : 1
21/04/30 17:00:45 INFO mapreduce.JobSubmitter: number of splits:2
21/04/30 17:00:45 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619788973438_0001
21/04/30 17:00:46 INFO impl.YarnClientImpl: Submitted application application_1619788973438_0001
21/04/30 17:00:46 INFO mapreduce.Job: The url to track the job: http://DESKTOP-SN0MSI7:8088/proxy/application_1619788973438
_0001/
21/04/30 17:00:46 INFO mapreduce.Job: Running job: job_1619788973438_0001
21/04/30 17:01:11 INFO mapreduce.Job: Job job_1619788973438_0001 running in uber mode : false
21/04/30 17:01:11 INFO mapreduce.Job:  map 0% reduce 0%
21/04/30 17:01:28 INFO mapreduce.Job:  map 100% reduce 0%
21/04/30 17:01:50 INFO mapreduce.Job:  map 100% reduce 100%
21/04/30 17:01:53 INFO mapreduce.Job: Job job_1619788973438_0001 completed successfully
21/04/30 17:01:53 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=17747
                FILE: Number of bytes written=406500
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=127591
                HDFS: Number of bytes written=661
```



Select Administrator: Command Prompt

```
                Map output materialized bytes=17753
                Input split bytes=236
                Combine input records=0
                Combine output records=0
                Reduce input groups=58
                Reduce shuffle bytes=17753
                Reduce input records=999
                Reduce output records=58
                Spilled Records=1998
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=180
                CPU time spent (ms)=4074
                Physical memory (bytes) snapshot=622325760
                Virtual memory (bytes) snapshot=834326528
                Total committed heap usage (bytes)=450363392
        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=127355
        File Output Format Counters
                Bytes Written=661

C:\Users\aaziz>
```

Once done, we can view and download our output result file to our output directory on hadoop.


File information - part-00000

Download

Block information -- Block 0

Block ID: 1073741832

Block Pool ID: BP-20745571-192.168.1.39-1619648186150

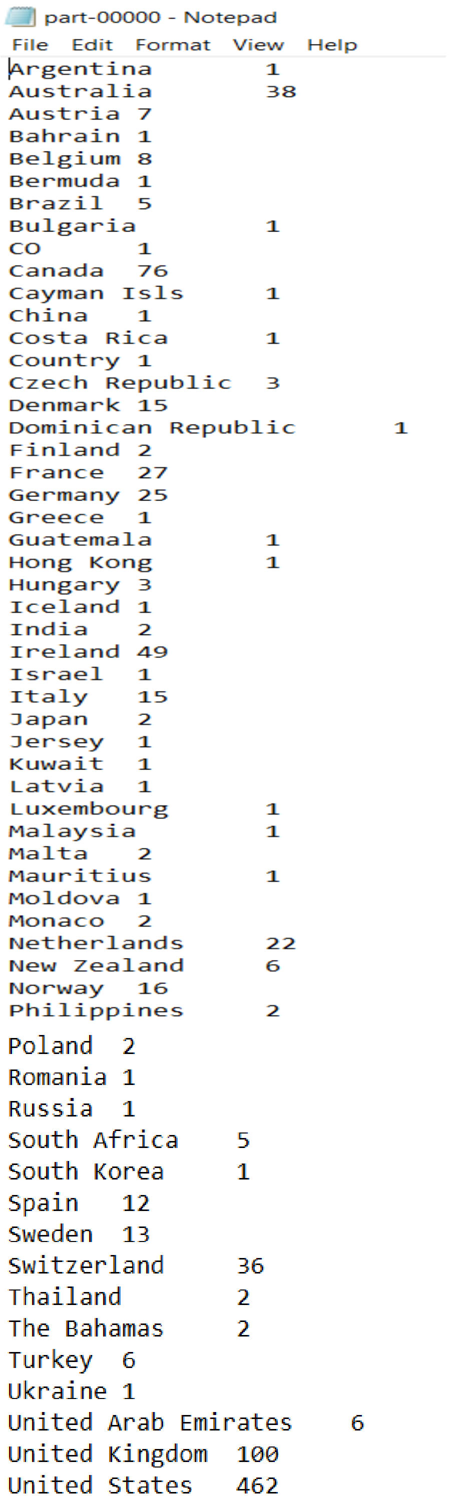
Generation Stamp: 1008

Size: 661

Availability:

- DESKTOP-SN0MSI7

9- Output result



```
Argentina       1
Australia       38
Austria 7
Bahrain 1
Belgium 8
Bermuda 1
Brazil  5
Bulgaria        1
CO      1
Canada  76
Cayman Isls     1
China   1
Costa Rica      1
Country 1
Czech Republic  3
Denmark 15
Dominican Republic      1
Finland 2
France  27
Germany 25
Greece  1
Guatemala       1
Hong Kong       1
Hungary 3
Iceland 1
India   2
Ireland 49
Israel  1
Italy   15
Japan   2
Jersey  1
Kuwait  1
Latvia  1
Luxembourg      1
Malaysia        1
Malta   2
Mauritius       1
Moldova 1
Monaco  2
Netherlands     22
New Zealand     6
Norway  16
Philippines     2
Poland  2
Romania 1
Russia  1
South Africa    5
South Korea     1
Spain   12
Sweden  13
Switzerland     36
Thailand        2
The Bahamas     2
Turkey  6
Ukraine 1
United Arab Emirates    6
United Kingdom  100
United States   462
```