G. H. Raisoni Institute of Engineering & Management, Pune

DEPARTMENT: ARTIFICAL INTELLIGIENCE

ACADEMIC YEAR: 2022-23

CLASS: T.Y. BTech SEMESTER: VI

Subject Name: BIG DATA COMPUTING

LAB MANUAL

Sr. No.	Name of the Experiments
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Assignment 1

Title: To install Hadoop framework, configure it and setup a single node cluster. Use web based tools to monitors your Hadoop setup.

SOFTWARE REQUIREMENTS:
1.Ubuntu

2. jdk 8

THEORY:
Installation of Hadoop:
Hadoop software can be installed in three modes of operation

• Stand Alone Mode: Hadoop is a distributed software and is designed to run on a commodity of machines. However, we can install it on a single node in stand-alone mode. In this mode, Hadoop software runs as a single monolithic java process. This mode is extremely useful for debugging purpose. You can first test run your Map-Reduce application in this mode on small data, before actually executing it on cluster with big data.

• Pseudo Distributed Mode: In this mode also, Hadoop software is installed on a Single Node. Various daemons of Hadoop will run on the same machine as separate java processes. Hence all the daemons namely NameNode, DataNode, SecondaryNameNode, JobTracker, TaskTracker run on single machine.

• Fully Distributed Mode: In Fully Distributed Mode, the daemons NameNode, JobTracker, SecondaryNameNode (Optional and can be run on a separate node) run on the Master Node. The daemons DataNode and TaskTracker run on the Slave Node.

.....Single Node Installation...........

```
sudo apt-get update

ssh-keygen

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

----------**downloading and installing java (Hadoop requires a working Java 1.6+)**--------

```
sudo apt-get install openjdk-8-jdk -y


-----------------------------**downloading hadoop**------------------------

wget https://archive.apache.org/dist/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz

tar -xzvf hadoop-1.2.1.tar.gz

sudo mv hadoop-1.2.1 /usr/local/hadoop

---------------------------**configuring bashrc linux env setup**---------------

#Configuring bashrc linux env

nano ~/.bashrc

export HADOOP_PREFIX=/usr/local/hadoop/
export PATH=$PATH:$HADOOP_PREFIX/bin
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME

#Setting hadoop env

nano /usr/local/hadoop/conf/hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_OPTS=-Djava.net.preferIPV4Stack=true

#Configuring core-site.xml

nano /usr/local/hadoop/conf/core-site.xml

<property>
```

```
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>

<property>
<name>hadoop.tmp.dir</name>
<value>/usr/local/hadoop/tmp</value>
</property>
```

#Configuring hdfs-site.xml

```
nano /usr/local/hadoop/conf/hdfs-site.xml
```

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
```

#Configuring mapred-site.xml

```
nano /usr/local/hadoop/conf/mapred-site.xml
```

```
<property>
<name>mapred.job.tracker</name>
<value>hdfs://localhost:9001</value>
</property>
```

----------**making tmp dir**---------

```
mkdir /usr/local/hadoop/tmp
```

-------------**exec bash**-------

exec bash

----------------**formatting hadoop namenode**--------------

hadoop namenode -format

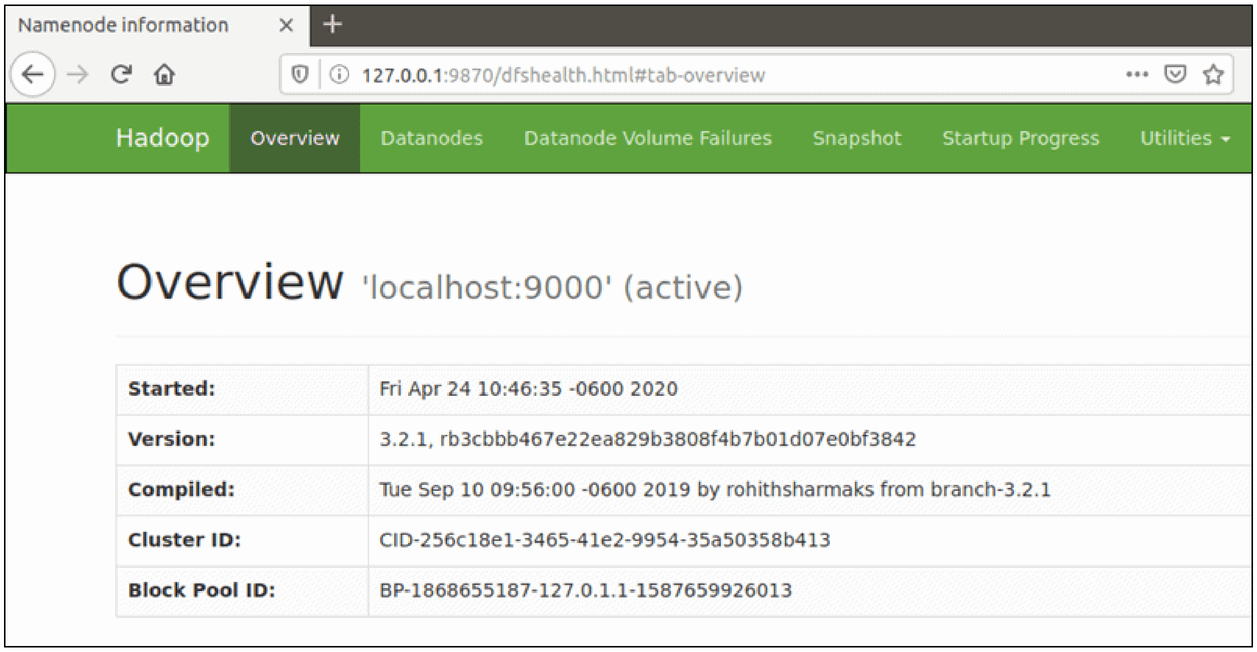
---------------**starting hadoop services(daemons)**-------------------

start-dfs.sh
start-mapred.sh
jps

Access Hadoop UI from Browser

http://localhost:9870

The NameNode user interface provides a comprehensive overview of the entire cluster.

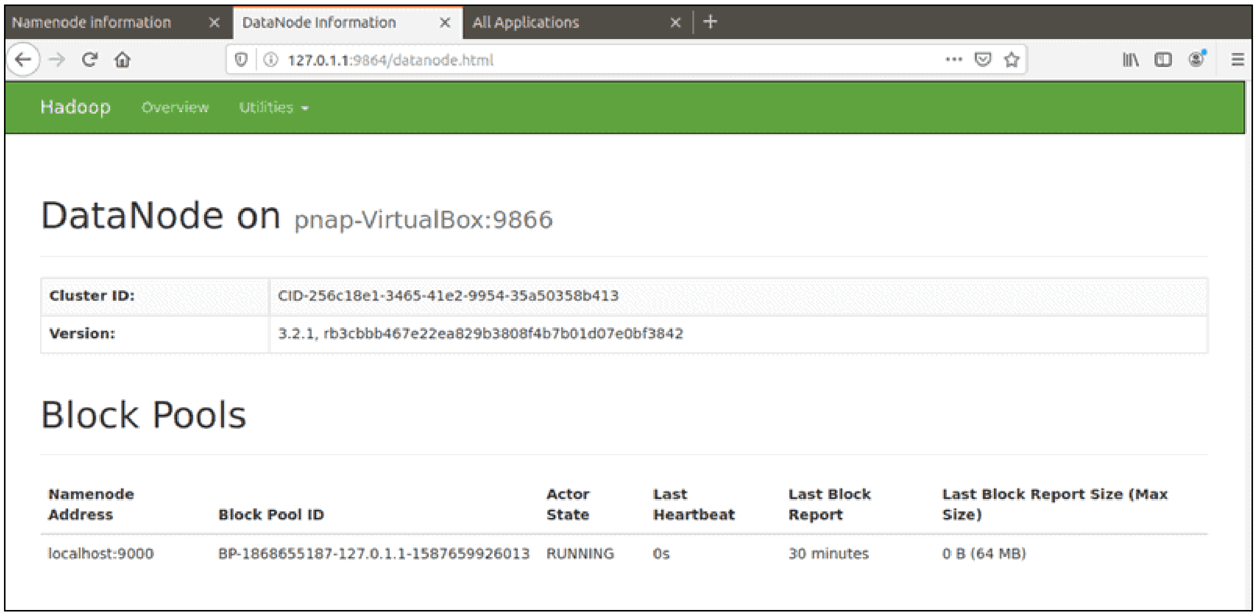


Overview 'localhost:9000' (active)

Started:	Fri Apr 24 10:46:35 -0600 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 09:56:00 -0600 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-256c18e1-3465-41e2-9954-35a50358b413
Block Pool ID:	BP-1868655187-127.0.1.1-1587659926013

The default port **9864** is used to access individual DataNodes directly from your browser:

```
http://localhost:9864
```



DataNode on pnap-VirtualBox:9866

Cluster ID:	CID-256c18e1-3465-41e2-9954-35a50358b413
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842

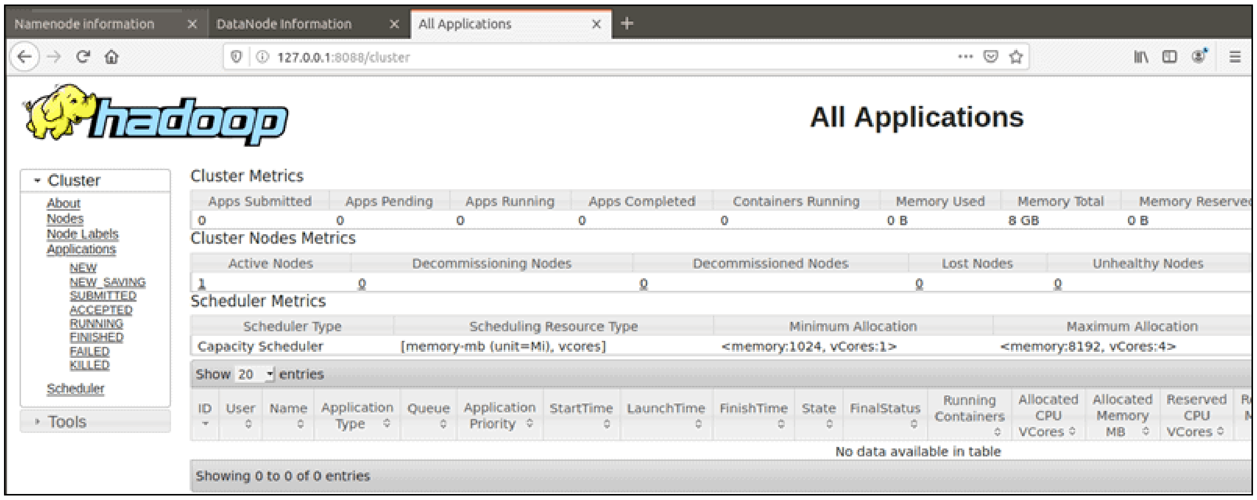
Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	BP-1868655187-127.0.1.1-1587659926013	RUNNING	0s	30 minutes	0 B (64 MB)

The YARN Resource Manager is accessible on port **8088**:

```
http://localhost:8088
```

The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved
0	0	0	0	0	0 B	8 GB	0 B

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores
No data available in table														

Showing 0 to 0 of 0 entries

Conclusion :

Assignment 2

Title: To implement file management task in Hadoop HDFS like adding, retriving, and deleting files.

SOFTWARE REQUIREMENTS:

1.Ubuntu

2. jdk8

THEORY:

Implement the following file management tasks in Hadoop:

i. Adding files and directories

ii. Retrieving files
iii. Deleting files

<u>Adding Files and Directories to HDFS</u>

```
hadoop fs -mkdir /user/chuck
```

```
hadoop fs -put
```

```
hadoop fs -put example.txt /user/chuck
```

<u>Retrieving Files from HDFS</u>

The Hadoop command get copies files from HDFS back to the local filesystem. To retrieve example.txt, we can run the following command:

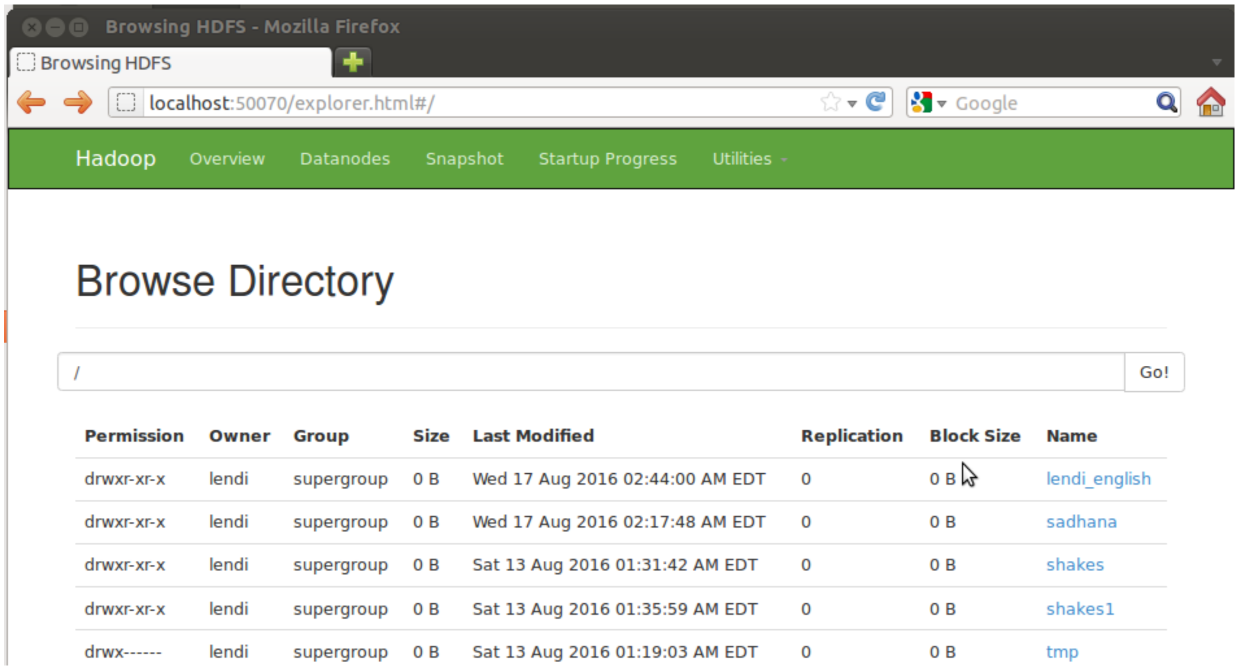
```
hadoop fs -cat

example.txt
```

<u>Deleting files from HDFS</u>

```
hadoop fs -rm example.txt
```

- Command for creating a directory in hdfs is “hdfs dfs –mkdir /lendicse”.
- Adding directory is done through the command “hdfs dfs –put lendi_english /”.

4.1 INPUT/OUTPUT:



Browse Directory

/

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:44:00 AM EDT	0	0 B	lendi_english
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:17:48 AM EDT	0	0 B	sadhana
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:31:42 AM EDT	0	0 B	shakes
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:35:59 AM EDT	0	0 B	shakes1
drwx------	lendi	supergroup	0 B	Sat 13 Aug 2016 01:19:03 AM EDT	0	0 B	tmp

Assignment 3

Title: To implement a word count application using the MapReduce API.

SOFTWARE REQUIREMENTS:

1.Ubuntu

2. jdk 8

THEORY:

WordCount is a simple program which counts the number of occurrences of each word in a given text input data set. WordCount fits very well with the MapReduce programming model making it a great example to understand the Hadoop Map/Reduce programming style. Our implementation consists of three main parts:

1. Mapper
2. Reducer
3. Driver

Step-1. Write a Mapper

A Mapper overrides the —map‖ function from the Class "org.apache.hadoop.mapreduce.Mapper" which provides <key, value> pairs as the input. A Mapper implementation may output

<key,value> pairs using the provided Context .

Input value of the WordCount Map task will be a line of text from the input data file and the key would be the line number <line_number, line_of_text> . Map task outputs <word, one> for each word in the line of text.

Pseudo-code

```
void Map (key, value){
        for each word x in
            value:
```

```
output.collect(x,1);
```

Step-2. Write a Reducer

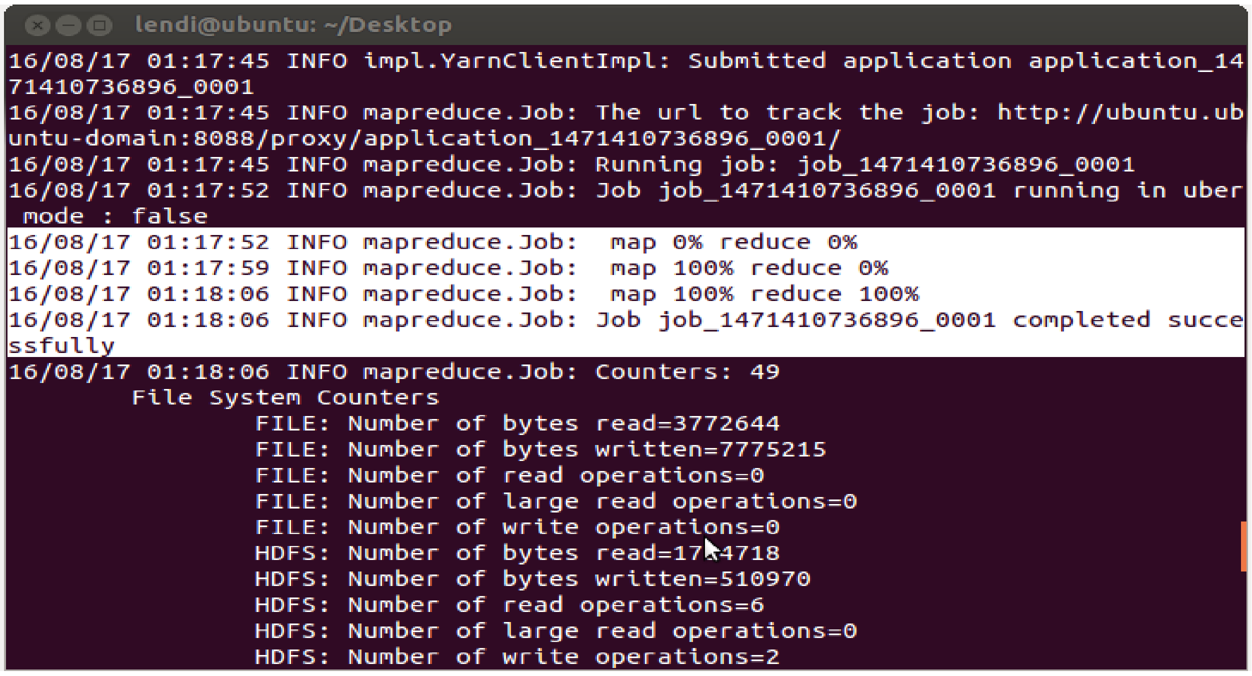
A Reducer collects the intermediate <key,value> output from multiple map tasks and assemble a single result. Here, the WordCount program will sum up the occurrence of each word to pairs as

<word, occurrence>.

Pseudo-code

```
void Reduce (keyword, <list of value>){ for
    each x in <list of value>:
        sum+=x;
    final_output.collect(keyword,
    sum);
}
```

5.2 INPUT/OUTPUT:



```
lendi@ubuntu: ~/Desktop
16/08/17 01:17:45 INFO impl.YarnClientImpl: Submitted application application_14
71410736896_0001
16/08/17 01:17:45 INFO mapreduce.Job: The url to track the job: http://ubuntu.ub
untu-domain:8088/proxy/application_1471410736896_0001/
16/08/17 01:17:45 INFO mapreduce.Job: Running job: job_1471410736896_0001
16/08/17 01:17:52 INFO mapreduce.Job: Job job_1471410736896_0001 running in uber
 mode : false
16/08/17 01:17:52 INFO mapreduce.Job:  map 0% reduce 0%
16/08/17 01:17:59 INFO mapreduce.Job:  map 100% reduce 0%
16/08/17 01:18:06 INFO mapreduce.Job:  map 100% reduce 100%
16/08/17 01:18:06 INFO mapreduce.Job: Job job_1471410736896_0001 completed succe
ssfully
16/08/17 01:18:06 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=3772644
                FILE: Number of bytes written=7775215
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=17[illegible]4718
                HDFS: Number of bytes written=510970
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

Assignment 7

PROGRAM LOGIC:

STEPS FOR INSTALLING APACHE PIG

1) Extract the pig-0.15.0.tar.gz and move to home directory
2) Set the environment of PIG in bashrc file.
3) Pig can run in two

modes Local Mode and

Hadoop Mode Pig –x local and

pig

4) Grunt

Shell Grunt >

5) LOADING Data into Grunt Shell

DATA = LOAD <CLASSPATH> USING PigStorage(DELIMITER) as (ATTRIBUTE : DataType1, ATTRIBUTE : DataType2…..)

6) Describe

Data Describe

DATA;

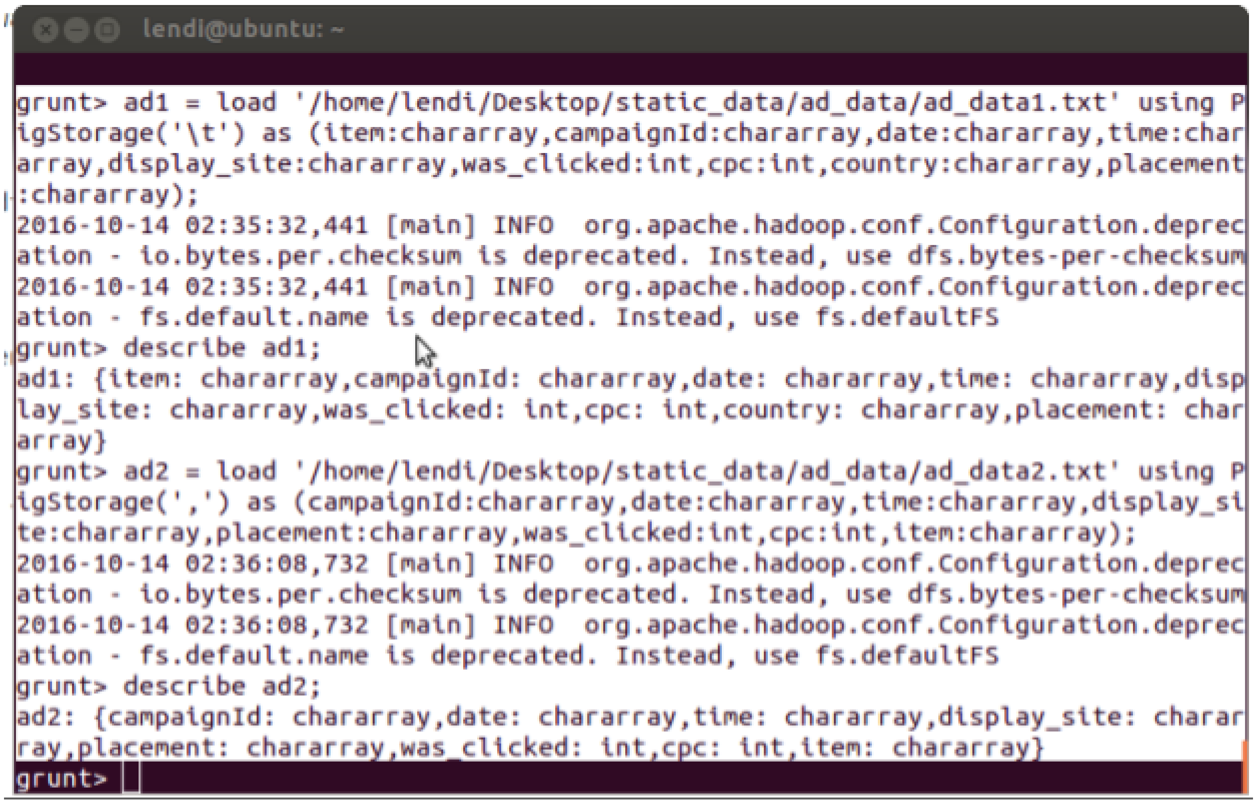
7) DUMP

Data Dump

DATA;

8.1 INPUT/OUTPUT:

Input as Website Click Count Data



```
grunt> ad1 = load '/home/lendi/Desktop/static_data/ad_data/ad_data1.txt' using P
igStorage('\t') as (item:chararray,campaignId:chararray,date:chararray,time:char
array,display_site:chararray,was_clicked:int,cpc:int,country:chararray,placement
:chararray);
2016-10-14 02:35:32,441 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2016-10-14 02:35:32,441 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> describe ad1;
ad1: {item: chararray,campaignId: chararray,date: chararray,time: chararray,disp
lay_site: chararray,was_clicked: int,cpc: int,country: chararray,placement: char
array}
grunt> ad2 = load '/home/lendi/Desktop/static_data/ad_data/ad_data2.txt' using P
igStorage(',') as (campaignId:chararray,date:chararray,time:chararray,display_si
te:chararray,placement:chararray,was_clicked:int,cpc:int,item:chararray);
2016-10-14 02:36:08,732 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2016-10-14 02:36:08,732 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> describe ad2;
ad2: {campaignId: chararray,date: chararray,time: chararray,display_site: charar
ray,placement: chararray,was_clicked: int,cpc: int,item: chararray}
grunt>
```

PROGRAM LOGIC:

FILTER Data

FDATA = FILTER DATA by ATTRIBUTE = VALUE;

GROUP Data

GDATA = GROUP DATA by ATTRIBUTE;

Iterating Data

FOR_DATA = FOREACH DATA GENERATE GROUP AS GROUP_FUN,
ATTRIBUTE = <VALUE>

Sorting Data

SORT_DATA = ORDER DATA BY ATTRIBUTE WITH CONDITION;

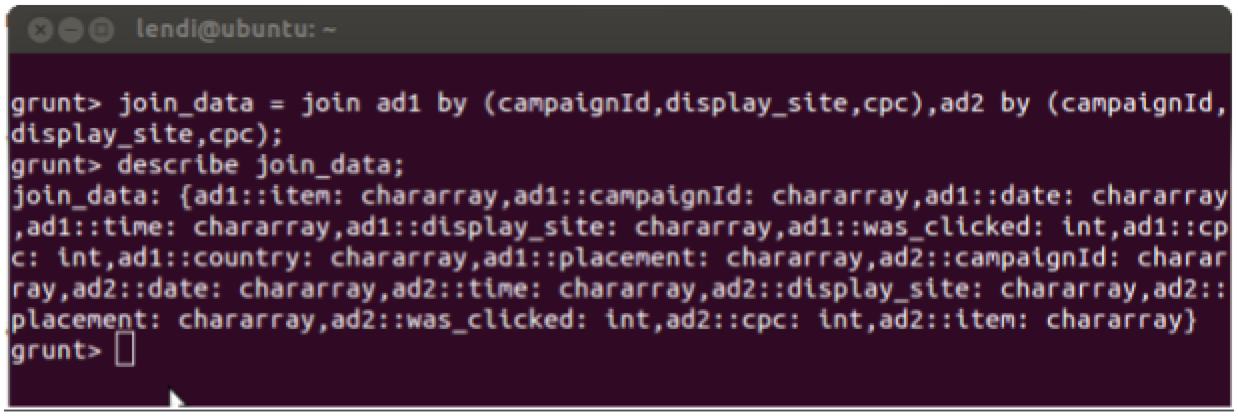
LIMIT Data

LIMIT_DATA = LIMIT DATA COUNT;

JOIN Data

JOIN DATA1 BY (ATTRIBUTE1,ATTRIBUTE2….) , DATA2 BY (ATTRIBUTE3,ATTRIBUTE….N)

9.1 INPUT / OUTPUT :



```
grunt> join_data = join ad1 by (campaignId,display_site,cpc),ad2 by (campaignId,
display_site,cpc);
grunt> describe join_data;
join_data: {ad1::item: chararray,ad1::campaignId: chararray,ad1::date: chararray
,ad1::time: chararray,ad1::display_site: chararray,ad1::was_clicked: int,ad1::cp
c: int,ad1::country: chararray,ad1::placement: chararray,ad2::campaignId: charar
ray,ad2::date: chararray,ad2::time: chararray,ad2::display_site: chararray,ad2::
placement: chararray,ad2::was_clicked: int,ad2::cpc: int,ad2::item: chararray}
grunt> 
```

EXPERIMENT NO.4

Creating the HDFS tables and loading them in Hive

HIVE tables provide us the schema to store data in various formats (like CSV). Hive provides multiple ways to add data to the tables. We can use DML(Data Manipulation Language) queries in Hive to import or add data to the table. One can also directly put the table into the hive with **HDFS commands**. In case we have data in Relational Databases like MySQL, ORACLE, IBM DB2, etc. then we can use ***Sqoop*** to efficiently transfer PetaBytes of data between Hadoop and Hive. In this particular tutorial, we will be using Hive DML queries to Load or INSERT data to the Hive table.
To perform the below operation make sure your hive is running. Below are the steps to launch a hive on your local system.

Step 1: Start all your Hadoop Daemon

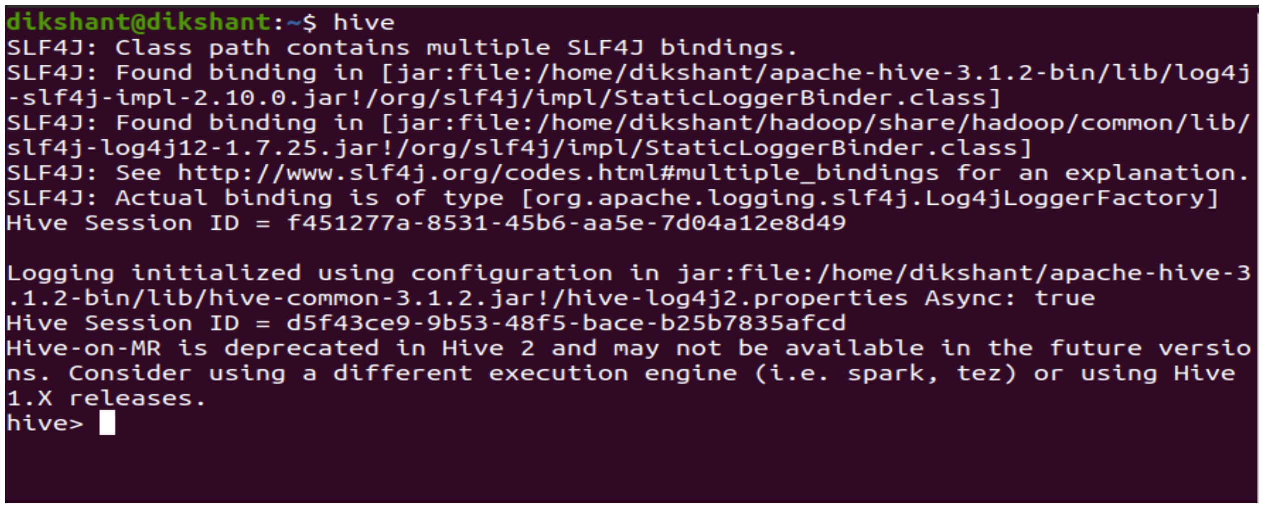
start-dfs.sh # this will start namenode, datanode and secondary namenode

start-yarn.sh # this will start node manager and resource manager

jps # To check running daemons

Step 2: Launch hive from terminal

hive



```
dikshant@dikshant:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/dikshant/apache-hive-3.1.2-bin/lib/log4j
-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/dikshant/hadoop/share/hadoop/common/lib/
slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = f451277a-8531-45b6-aa5e-7d04a12e8d49

Logging initialized using configuration in jar:file:/home/dikshant/apache-hive-3
.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = d5f43ce9-9b53-48f5-bace-b25b7835afcd
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
hive> 
```

In hive with DML statements, we **can add data to the Hive table in 2 different ways**.

- Using INSERT Command
- Load Data Statement

1. Using INSERT Command

Syntax:
INSERT INTO TABLE <table_name> VALUES (<add values as per column entity>);

Example:
To insert data into the table let's create a table with the name ***student*** (By default hive uses its *default* database to store hive tables).
Command:
CREATE TABLE IF NOT EXISTS student(

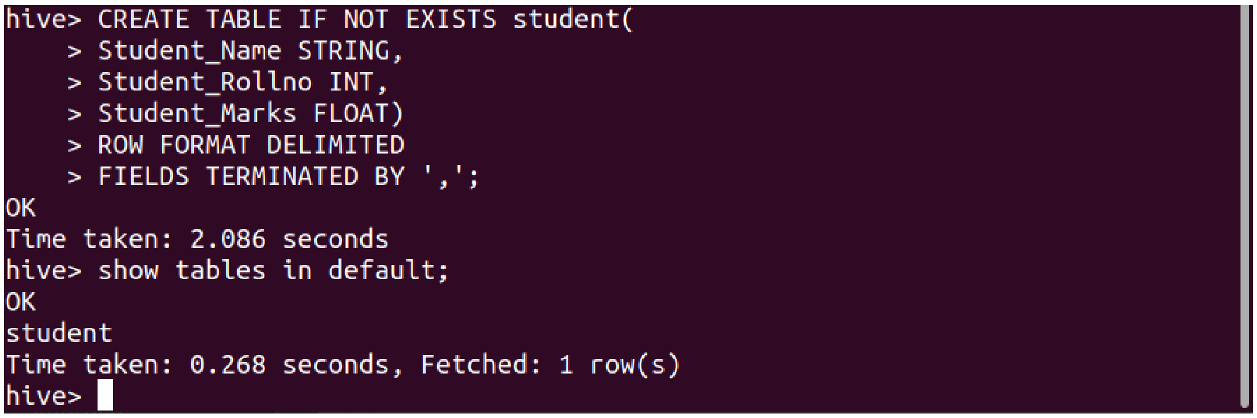
Student_Name STRING,

Student_Rollno INT,

Student_Marks FLOAT)

ROW FORMAT DELIMITED

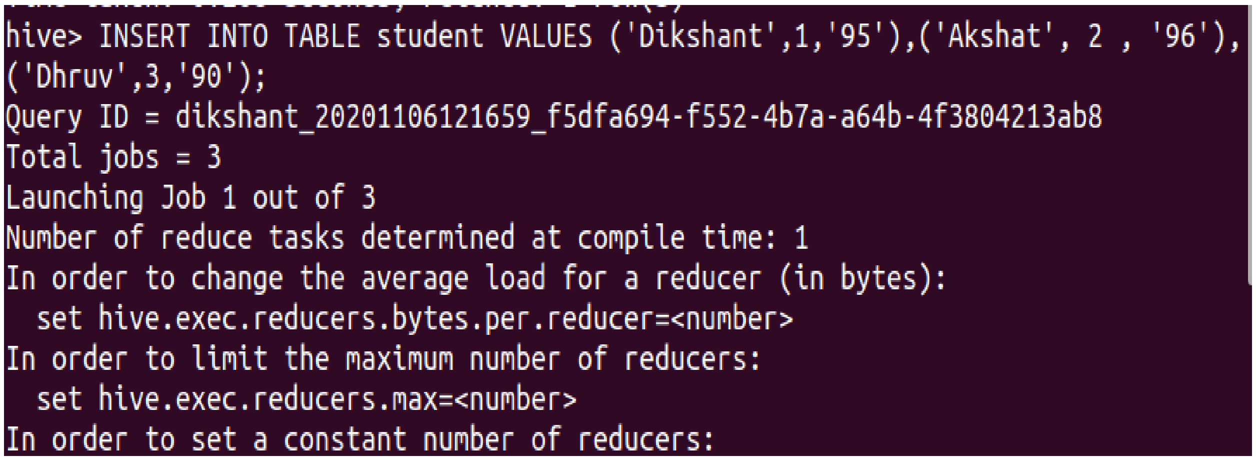
FIELDS TERMINATED BY ',';



```
hive> CREATE TABLE IF NOT EXISTS student(
    > Student_Name STRING,
    > Student_Rollno INT,
    > Student_Marks FLOAT)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 2.086 seconds
hive> show tables in default;
OK
student
Time taken: 0.268 seconds, Fetched: 1 row(s)
hive> 
```

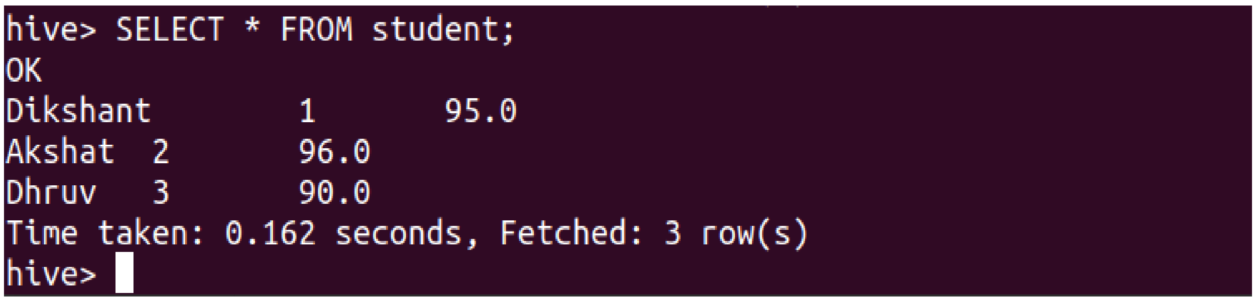
We have successfully created the *student* table in the Hive *default* database with the attribute *Student_Name*, *Student_Rollno*, and *Student_Marks* respectively.
Now, let's insert data into this table with an INSERT query.

INSERT Query:
INSERT INTO TABLE student VALUES ('Dikshant',1,'95'),('Akshat', 2 , '96'),('Dhruv',3,'90');



```
hive> INSERT INTO TABLE student VALUES ('Dikshant',1,'95'),('Akshat', 2 , '96'),
('Dhruv',3,'90');
Query ID = dikshant_20201106121659_f5dfa694-f552-4b7a-a64b-4f3804213ab8
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
```

We can check the data of the ***student*** table with the help of the below command.
SELECT * FROM student;



```
hive> SELECT * FROM student;
OK
Dikshant        1       95.0
Akshat  2       96.0
Dhruv   3       90.0
Time taken: 0.162 seconds, Fetched: 3 row(s)
hive> 
```

2. Load Data Statement

Hive provides us the functionality to load pre-created table entities either from our local file system or from HDFS. The *LOAD DATA* statement is used to load data into the hive table.

Syntax:

LOAD DATA [LOCAL] INPATH '<The table data location>' [OVERWRITE] INTO TABLE <table_name>;

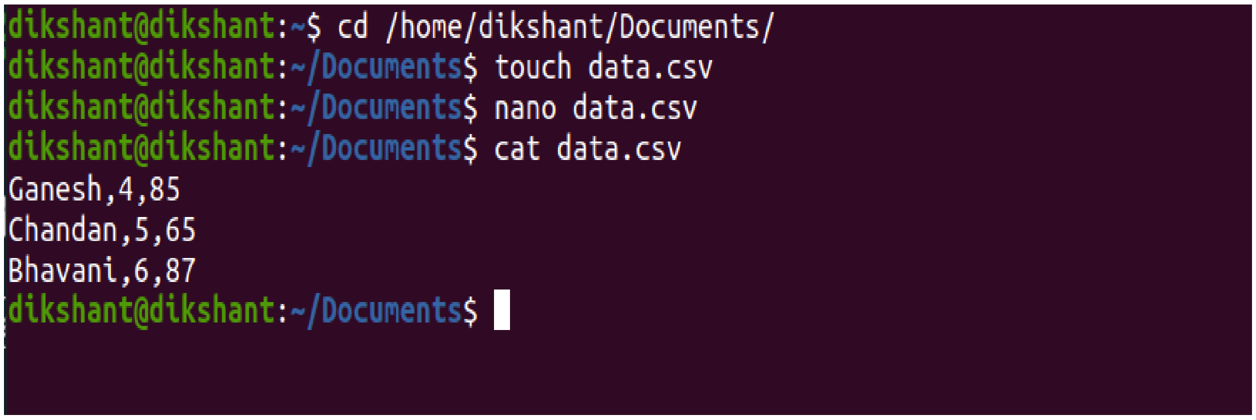
Note:

- The *LOCAL* Switch specifies that the data we are loading is available in our Local File System. If the *LOCAL* switch is not used, the hive will consider the location as an HDFS path location.
- The OVERWRITE switch allows us to overwrite the table data.

Let's make a CSV(Comma Separated Values) file with the name *data.csv* since we have provided ',' as a field terminator while creating a table in the hive. We are creating this file in our local file system at '*/home/dikshant/Documents*' for demonstration purposes.

Command:

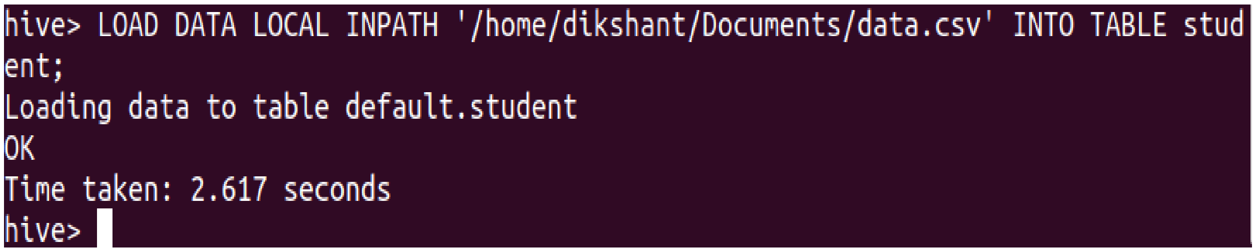
```
cd /home/dikshant/Documents    // To change the directory

touch data.csv                 // use to create data.csv file

nano data.csv                  // nano is a linux command line editor to edit files

cat data.csv                 // cat is used to see content of file
```



```
dikshant@dikshant:~$ cd /home/dikshant/Documents/
dikshant@dikshant:~/Documents$ touch data.csv
dikshant@dikshant:~/Documents$ nano data.csv
dikshant@dikshant:~/Documents$ cat data.csv
Ganesh,4,85
Chandan,5,65
Bhavani,6,87
dikshant@dikshant:~/Documents$ 
```

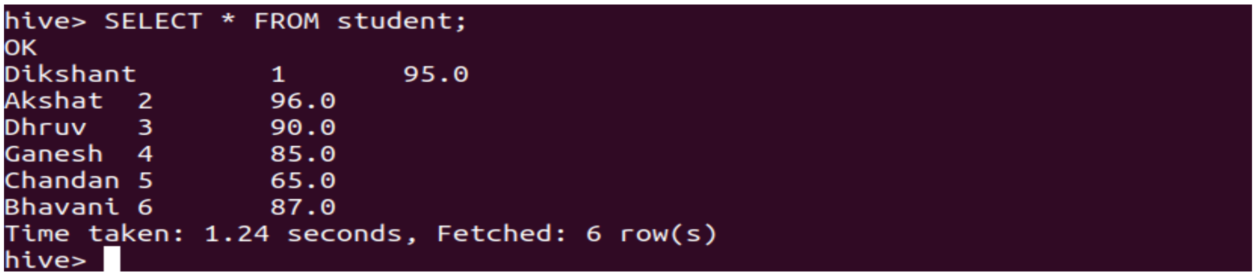
LOAD DATA to the student hive table with the help of the below command.

LOAD DATA LOCAL INPATH '/home/dikshant/Documents/data.csv' INTO TABLE student;



```
hive> LOAD DATA LOCAL INPATH '/home/dikshant/Documents/data.csv' INTO TABLE stud
ent;
Loading data to table default.student
OK
Time taken: 2.617 seconds
hive>
```

Let's see the *student* table content to observe the effect with the help of the below command.
SELECT * FROM student;



```
hive> SELECT * FROM student;
OK
Dikshant        1       95.0
Akshat  2       96.0
Dhruv   3       90.0
Ganesh  4       85.0
Chandan 5       65.0
Bhavani 6       87.0
Time taken: 1.24 seconds, Fetched: 6 row(s)
hive>
```

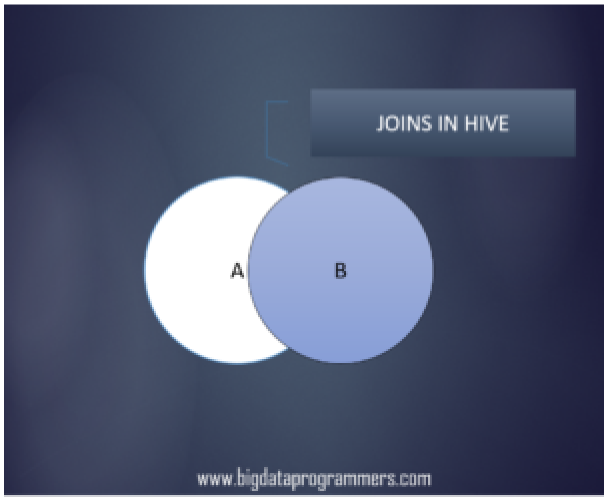
We can observe that we have successfully added the data to the *student* table.

EXPERIMENT-5

To create HDFS tables and load them in Hive and Implement joining of tables in Hive.

Requirement

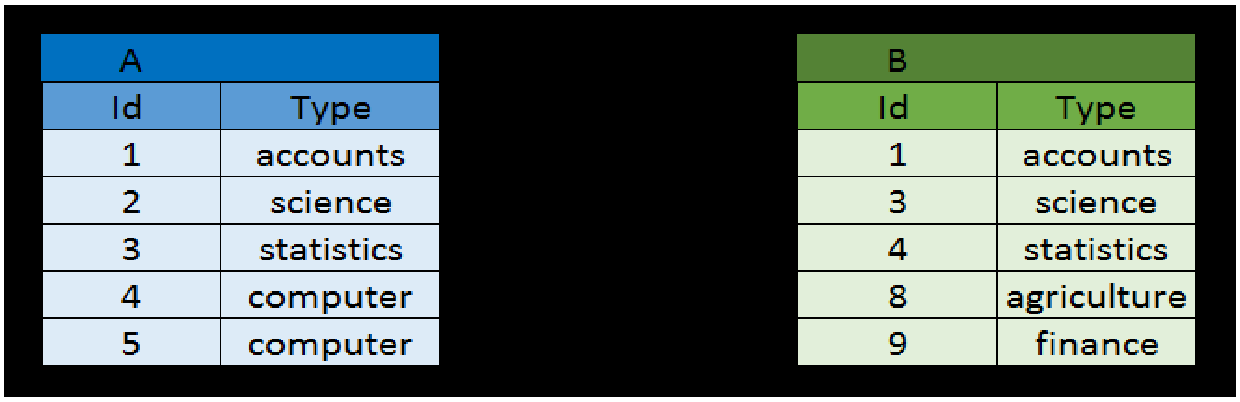
You have two table named as A and B. and you want to perform all types of join in hive. It will help you to understand, how join works in hive.



Solution

Step 1: Input Files

Download file Aand B from here. And place them into a local directory.File A and B are the comma delimited file, please refer below :-



A	
Id	Type
1	accounts
2	science
3	statistics
4	computer
5	computer

B	
Id	Type
1	accounts
3	science
4	statistics
8	agriculture
9	finance

I am placing these files into local directory ‘sample_files’

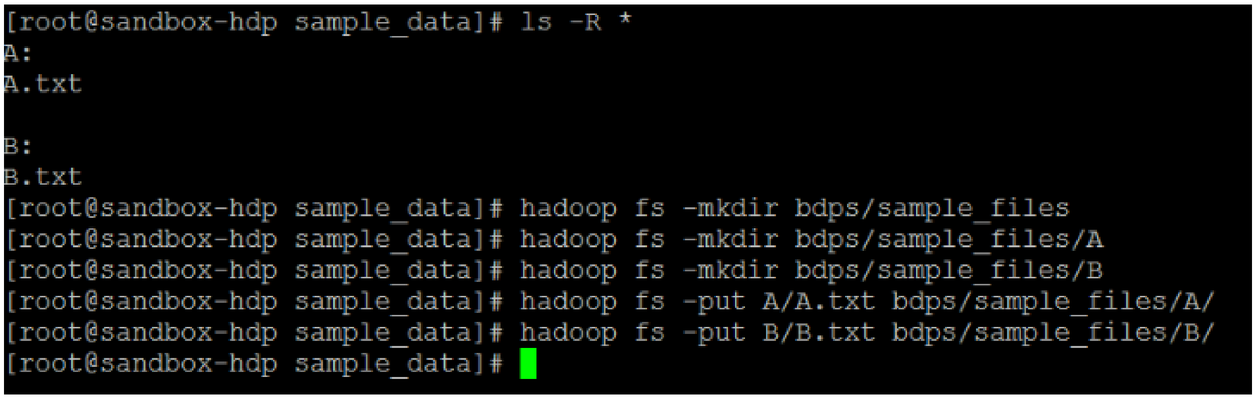
```
cd sample_files
```

```
ls -R *
```

Step 2: Loading the files into Hive.

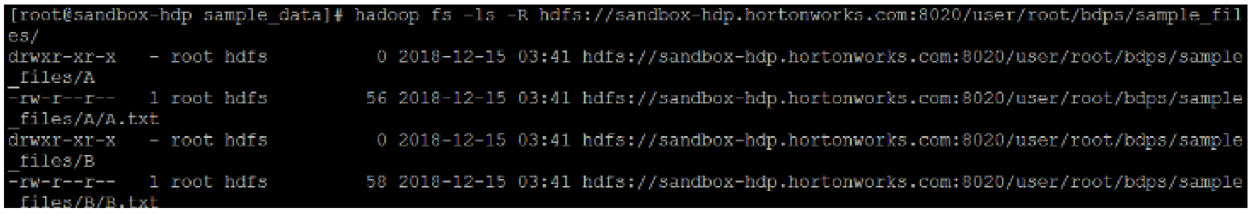
To load the files into hive,Let’s first put these files into hdfs location using below commands.

```
 hadoop fs -mkdir bdps/sample_files

hadoop fs -mkdir bdps/sample_files/A

hadoop fs -mkdir bdps/sample_files/B

hadoop fs -put A/A.txt bdps/sample_files/A/

hadoop fs -put B/B.txt bdps/sample_files/B/
```



```
[root@sandbox-hdp sample_data]# ls -R *
A:
A.txt

B:
B.txt
[root@sandbox-hdp sample_data]# hadoop fs -mkdir bdps/sample_files
[root@sandbox-hdp sample_data]# hadoop fs -mkdir bdps/sample_files/A
[root@sandbox-hdp sample_data]# hadoop fs -mkdir bdps/sample_files/B
[root@sandbox-hdp sample_data]# hadoop fs -put A/A.txt bdps/sample_files/A/
[root@sandbox-hdp sample_data]# hadoop fs -put B/B.txt bdps/sample_files/B/
[root@sandbox-hdp sample_data]#
```

you can check the files in hdfs using below command.



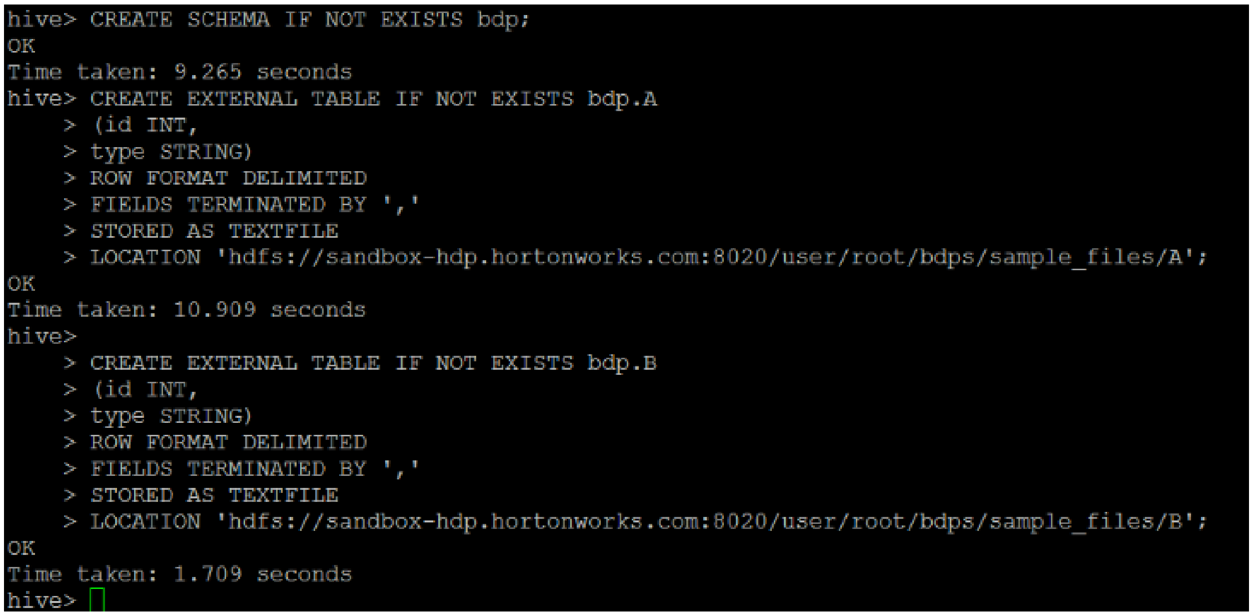
```
[root@sandbox-hdp sample_data]# hadoop fs -ls -R hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample_fil
es/
drwxr-xr-x   - root hdfs          0 2018-12-15 03:41 hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample
_files/A
-rw-r--r--   1 root hdfs         56 2018-12-15 03:41 hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample
_files/A/A.txt
drwxr-xr-x   - root hdfs          0 2018-12-15 03:41 hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample
_files/B
-rw-r--r--   1 root hdfs         58 2018-12-15 03:41 hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample
_files/B/B.txt
```

hadoop fs -ls -R hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample_files/

Now let’s create two hive table A and B for both the files,using below commands:-

```
 CREATE SCHEMA IF NOT EXISTS bdp;

CREATE EXTERNAL TABLE IF NOT EXISTS bdp.A(id INT,type STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'hdfs://sandboxhdp.hortonworks.com:8020/user/root/bdps/sample_files/A';


CREATE EXTERNAL TABLE IF NOT EXISTS bdp.B

(id INT,type STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE
```

LOCATION
'hdfs://sandboxhdp.hortonworks.com:8020/user/roo
t/bdps/sample_files/B';

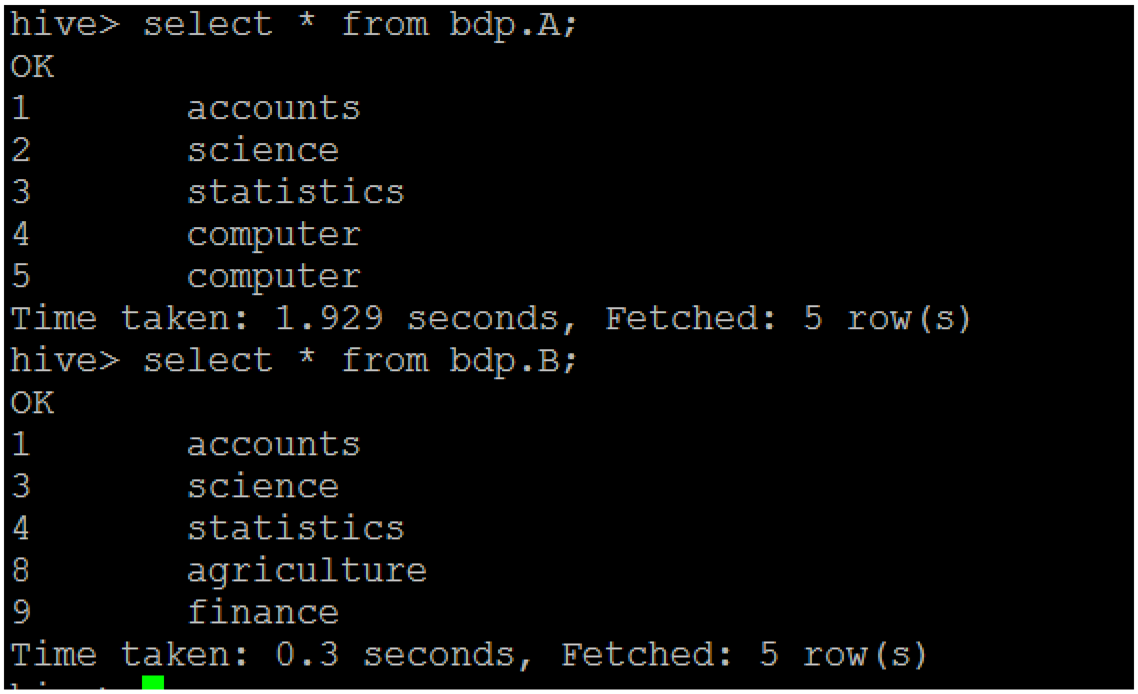


```
hive> CREATE SCHEMA IF NOT EXISTS bdp;
OK
Time taken: 9.265 seconds
hive> CREATE EXTERNAL TABLE IF NOT EXISTS bdp.A
    > (id INT,
    > type STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > LOCATION 'hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample_files/A';
OK
Time taken: 10.909 seconds
hive>
    > CREATE EXTERNAL TABLE IF NOT EXISTS bdp.B
    > (id INT,
    > type STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > LOCATION 'hdfs://sandbox-hdp.hortonworks.com:8020/user/root/bdps/sample_files/B';
OK
Time taken: 1.709 seconds
hive>
```

Let’s check whether data populated correctly or not using below commands :-

select * from bdp.A;

select * from bdp.B;



```
hive> select * from bdp.A;
OK
1       accounts
2       science
3       statistics
4       computer
5       computer
Time taken: 1.929 seconds, Fetched: 5 row(s)
hive> select * from bdp.B;
OK
1       accounts
3       science
4       statistics
8       agriculture
9       finance
Time taken: 0.3 seconds, Fetched: 5 row(s)
```

Let’s understand join one by one

A. Inner Join:

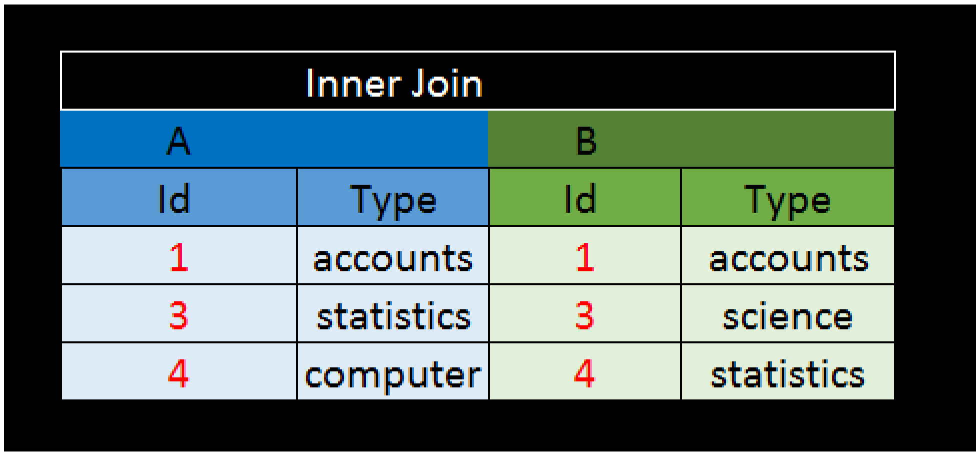
Sometimes it is required to have only common records out of two datasets. Now we have two table A & B, we are joining based on a key which is id.

So in output, only those records which match id with another dataset will come. Rest will be discarded.

Use below command to perform the inner join.

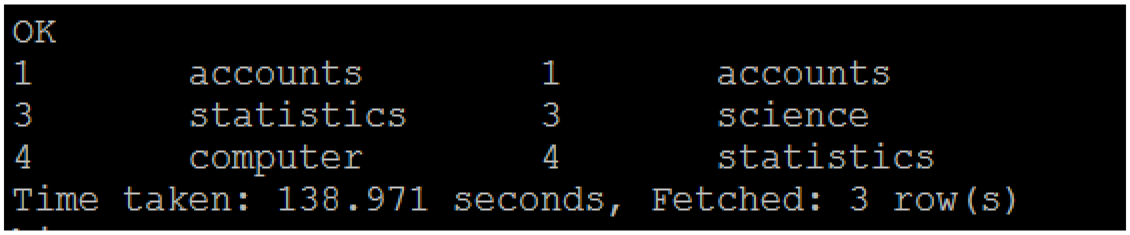
```
 select * from
(select * from bdp.A)T1
JOIN
(select * from bdp.B)T2
ON T1.id=T2.id;
```

Expected output:



Inner Join			
A		B	
Id	Type	Id	Type
1	accounts	1	accounts
3	statistics	3	science
4	computer	4	statistics

Please refer below screen shot for reference.



```
OK
1       accounts        1       accounts
3       statistics      3       science
4       computer        4       statistics
Time taken: 138.971 seconds, Fetched: 3 row(s)
```

As you can see only records which have the same id such as 1, 3, 4 are present in the output, rest have been discarded.

B. Left Join

this type of join is performed when we want to look up something from other datasets, the best example would be fetching a phone no of an employee from other datasets based on employee code.

Use below command to perform left join.

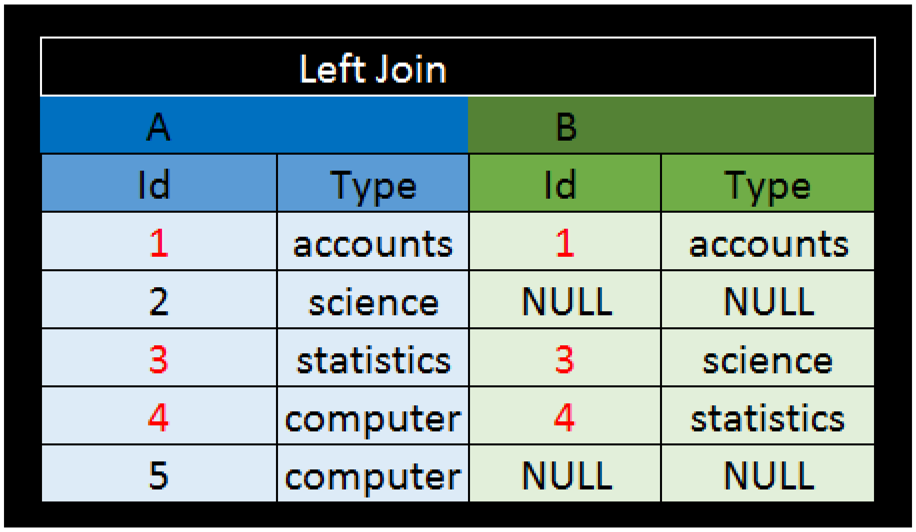
```
 select * from
(select * from bdp.A)T1
```

LEFT JOIN

(select * from bdp.B)T2

ON T1.id=T2.id;

Expected output



Left Join			
A		B	
Id	Type	Id	Type
1	accounts	1	accounts
2	science	NULL	NULL
3	statistics	3	science
4	computer	4	statistics
5	computer	NULL	NULL

Now we have all the records of left table A and matched records of table B.

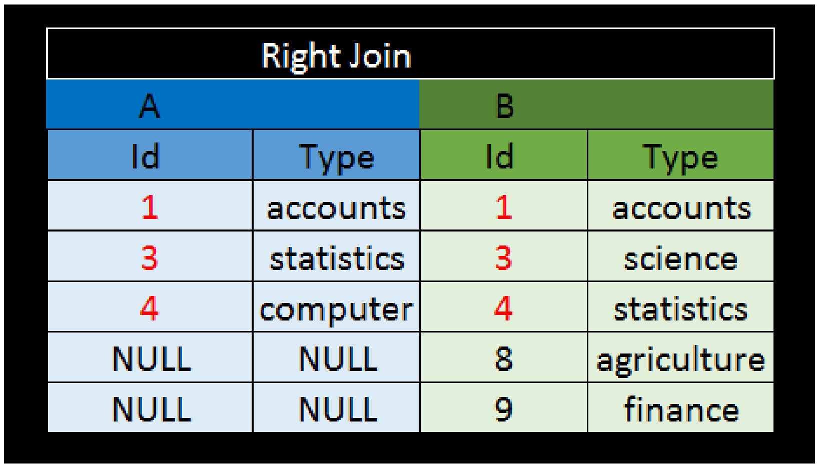
C. Right Join

This type of join is performed when we want to get all the data of look-up table with only matching records of left table.

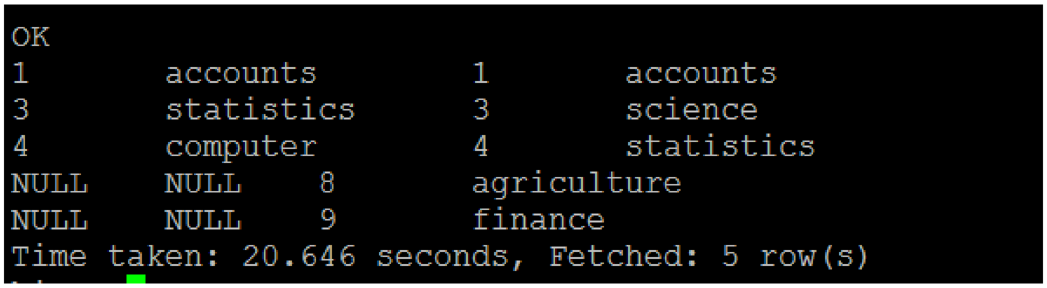
Use below command to perform right join.

```
 select * from
(select * from bdp.A)T1
RIGHT JOIN
(select * from bdp.B)T2
ON T1.id=T2.id;
```

Expected output



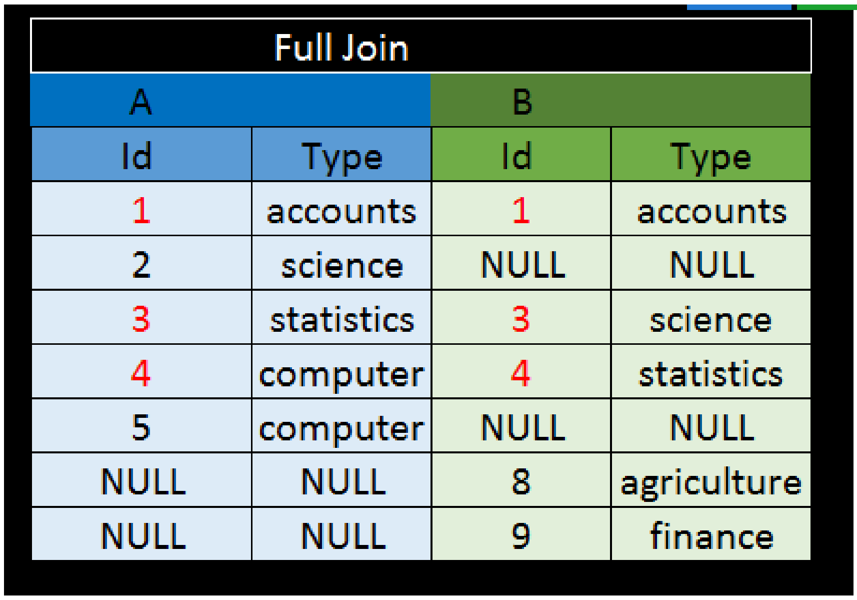
Right Join			
A		B	
Id	Type	Id	Type
1	accounts	1	accounts
3	statistics	3	science
4	computer	4	statistics
NULL	NULL	8	agriculture
NULL	NULL	9	finance



```
OK
1       accounts        1       accounts
3       statistics      3       science
4       computer        4       statistics
NULL    NULL    8       agriculture
NULL    NULL    9       finance
Time taken: 20.646 seconds, Fetched: 5 row(s)
```

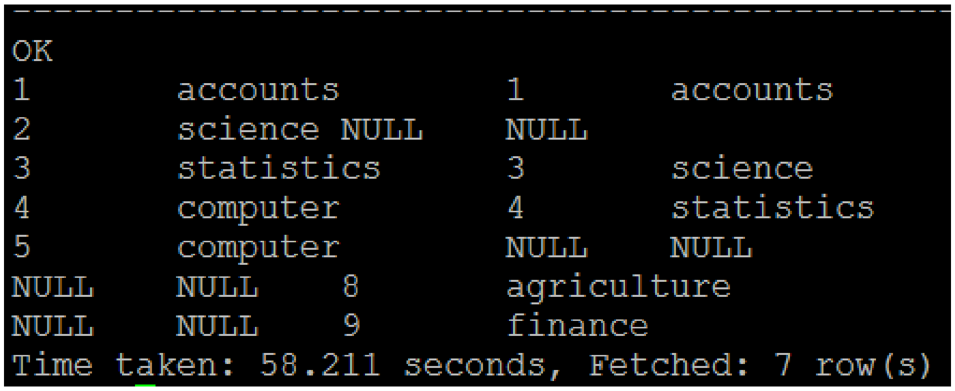
Now we have all the records of right table B and matched records of table A.

D.Full Join



Full Join			
A		B	
Id	Type	Id	Type
1	accounts	1	accounts
2	science	NULL	NULL
3	statistics	3	science
4	computer	4	statistics
5	computer	NULL	NULL
NULL	NULL	8	agriculture
NULL	NULL	9	finance

Now we have all matched and unmatched records in output as shown below.



```
OK
1       accounts        1       accounts
2       science NULL    NULL
3       statistics      3       science
4       computer        4       statistics
5       computer        NULL    NULL
NULL    NULL    8       agriculture
NULL    NULL    9       finance
Time taken: 58.211 seconds, Fetched: 7 row(s)
```

When it is needed to get all the matched and unmatched records out of two datasets, we can use full join. All data from left as well as from right datasets will appear in result set. Nonmatching records will have null have values in respective columns.

Use below command to perform full join.

```
 select * from
(select * from bdp.A)T1
FULL JOIN
(select * from bdp.B)T2
ON T1.id=T2.id;
```

Expected output

Now we have all matched and unmatched records in output as shown below.

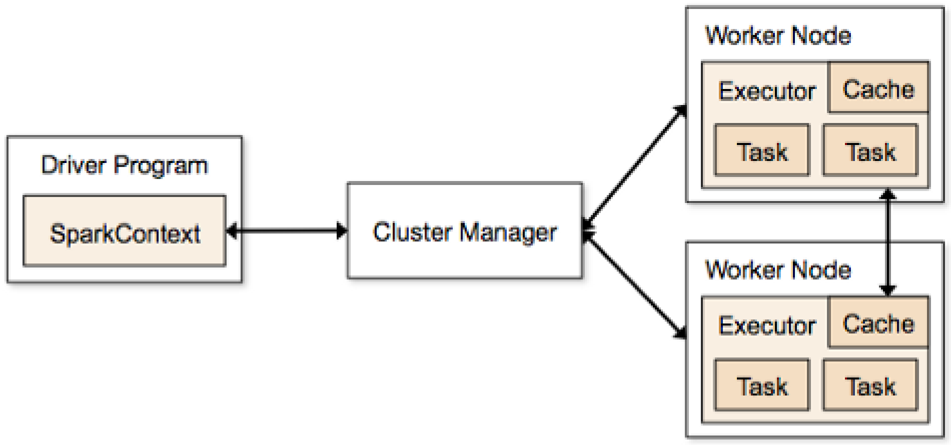
Wrapping Up

Joins are important when you have to deal with data which are present in more than a table. In real time we get files from many sources which have a relation between them, so to get meaningful information from these data-sets it needs to perform join to get combined result.

To install, deploy & configure Apache Spark Cluster. To Select the fields from the dataset using Spark SQL. To explore Spark shell and read from HDFS

Apache Spark follows a master/slave architecture with two main daemons and a cluster manager.

- Master Daemon — (Master/Driver Process)
- Worker Daemon –(Slave Process)
- Cluster Manager



A spark cluster has a single Master and any number of Slaves/Workers. The driver and the executors run their individual Java processes and users can run them on the same horizontal spark cluster or on separate machines.

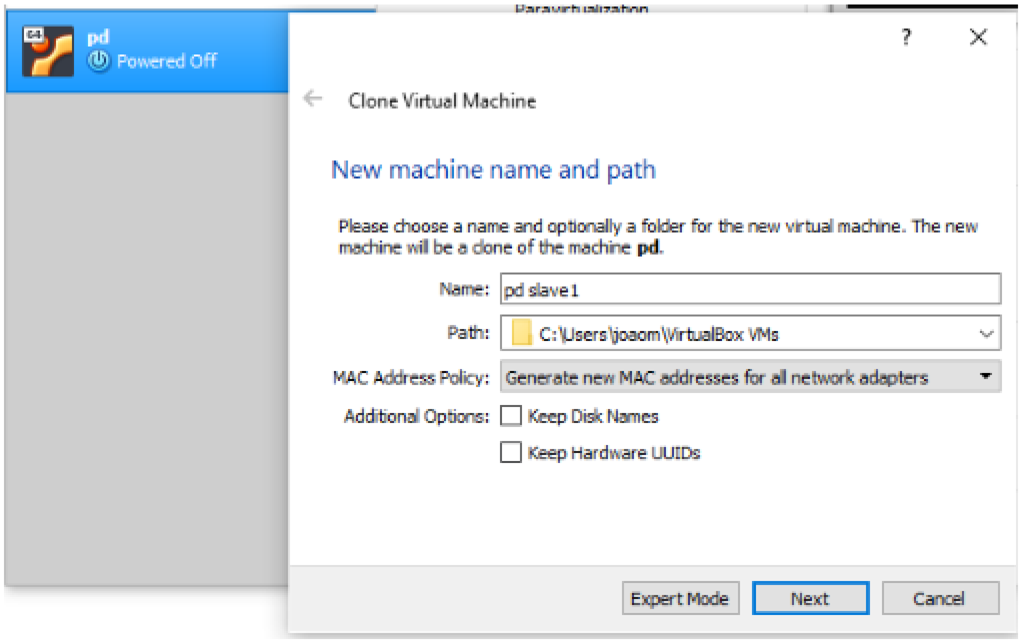
Pre-requirements

- Ubuntu 18.04 installed on a virtual machine.

1st Step:

Create 2 clones of the Virtual Machine you've previously created.

Make sure that you have the option "Generate new MAC addresses for all network adapters selected. Also, choose the option "Full Clone".



Please choose a name and optionally a folder for the new virtual machine. The new machine will be a clone of the machine **pd**.

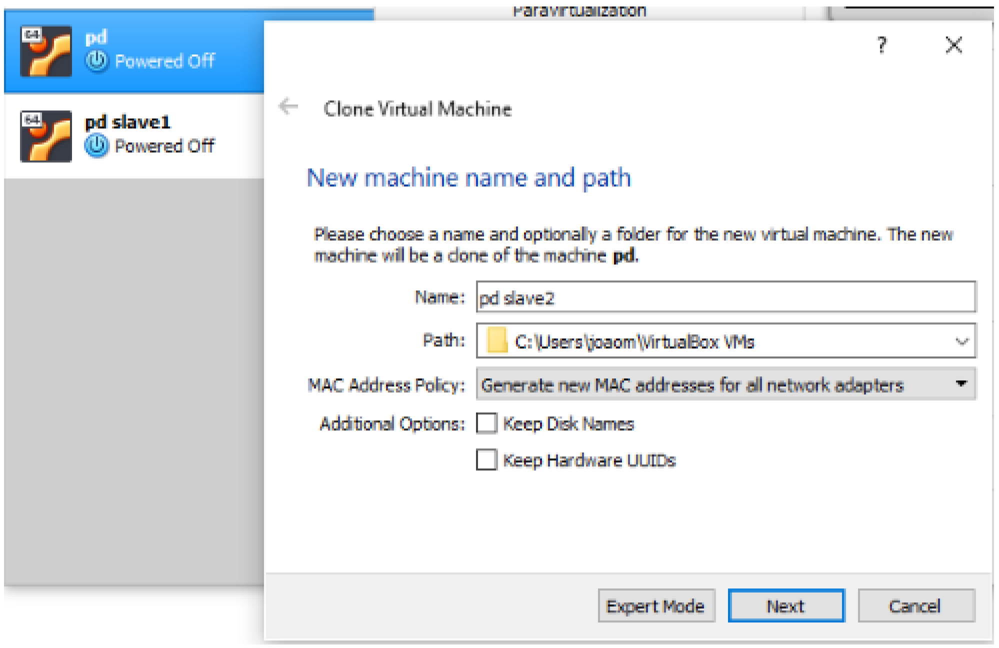
Name: pd slave1

Path: C:\Users\joaom\VirtualBox VMs

MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options: ☐ Keep Disk Names

☐ Keep Hardware UUIDs



New machine name and path

Please choose a name and optionally a folder for the new virtual machine. The new machine will be a clone of the machine **pd.**

Name: pd slave2

Path: C:\Users\joaom\VirtualBox VMs

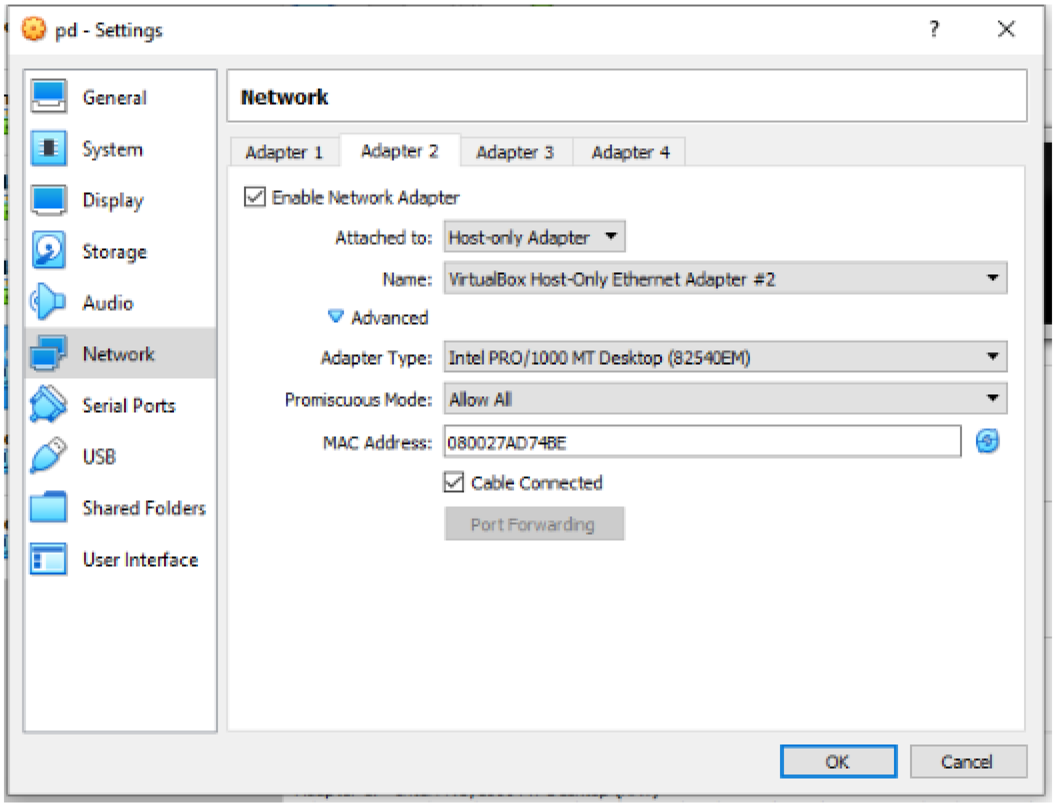
MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options: ☐ Keep Disk Names

☐ Keep Hardware UUIDs

2nd Step:

Make sure all the VM's have the following network configuration on Apapter 2:



☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter #2

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

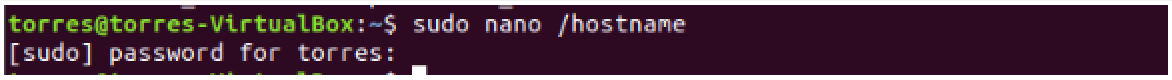
MAC Address: 080027AD74BE

☑ Cable Connected

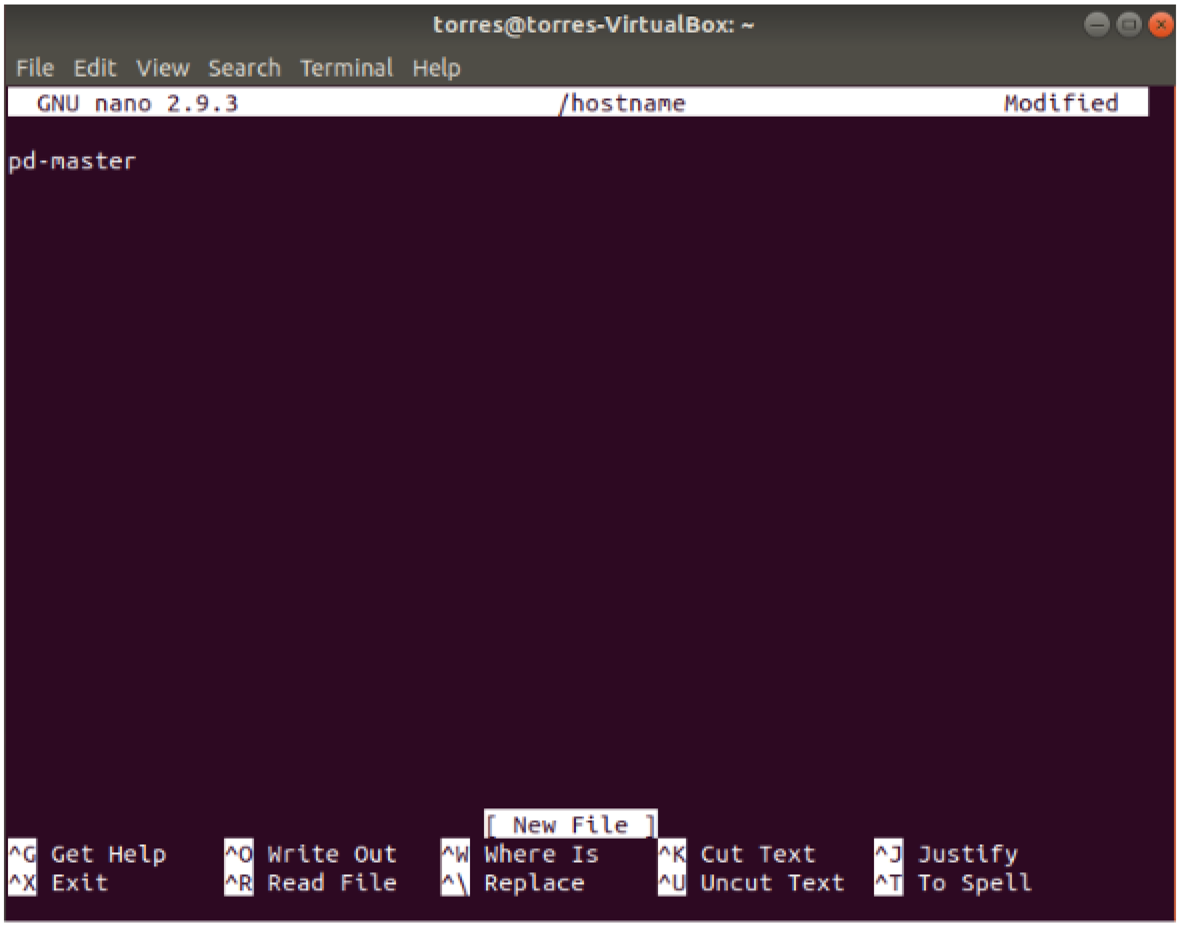
3rd Step:

Let’s change the hostname on each virtual machine. Open the file and type the name of the machina. Use this command:

```
sudo nano /hostname
```

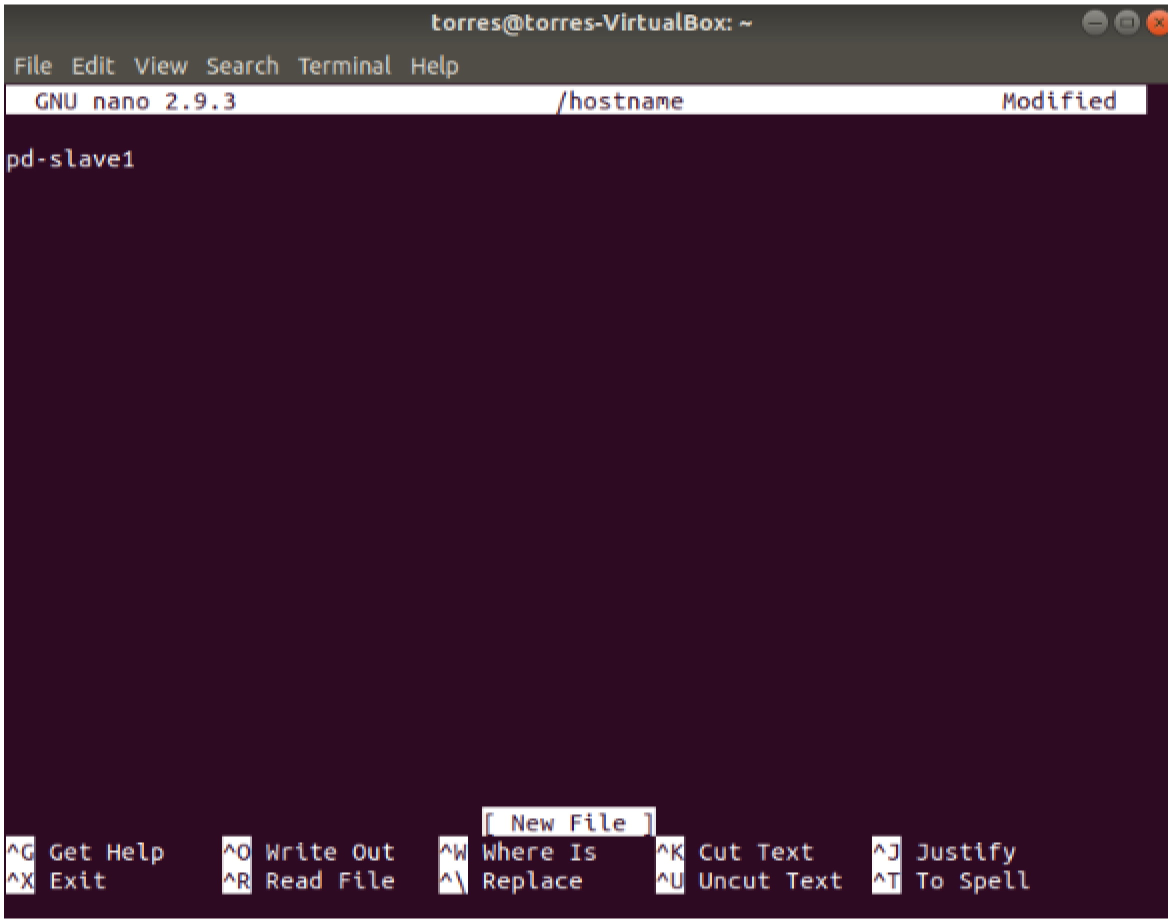


```
torres@torres-VirtualBox:~$ sudo nano /hostname
[sudo] password for torres:
```


torres@torres-VirtualBox: ~
File Edit View Search Terminal Help

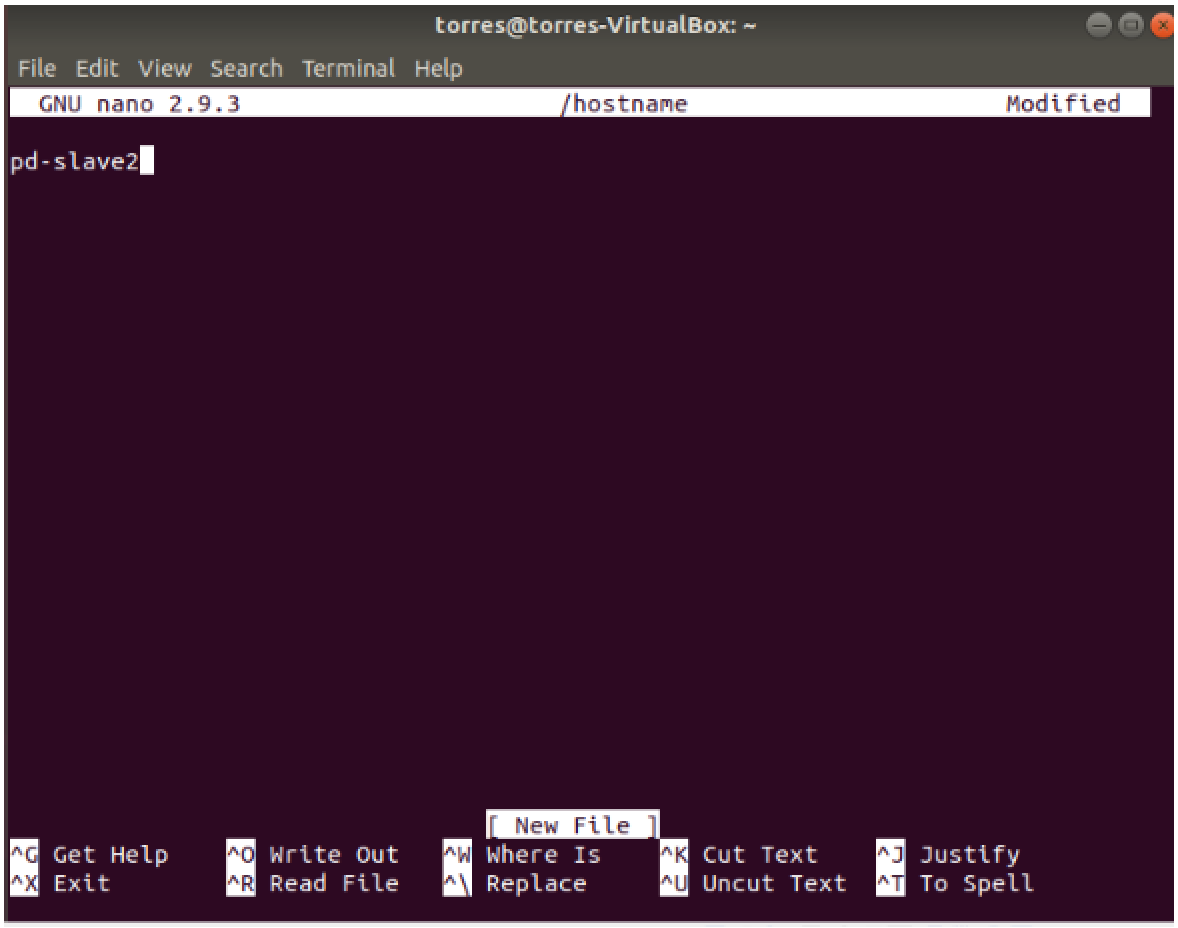
```
  GNU nano 2.9.3                /hostname                         Modified

pd-master
```

```
                                [ New File ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```


torres@torres-VirtualBox: ~
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                     /hostname                     Modified

pd-slave1
```

```
                                [ New File ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

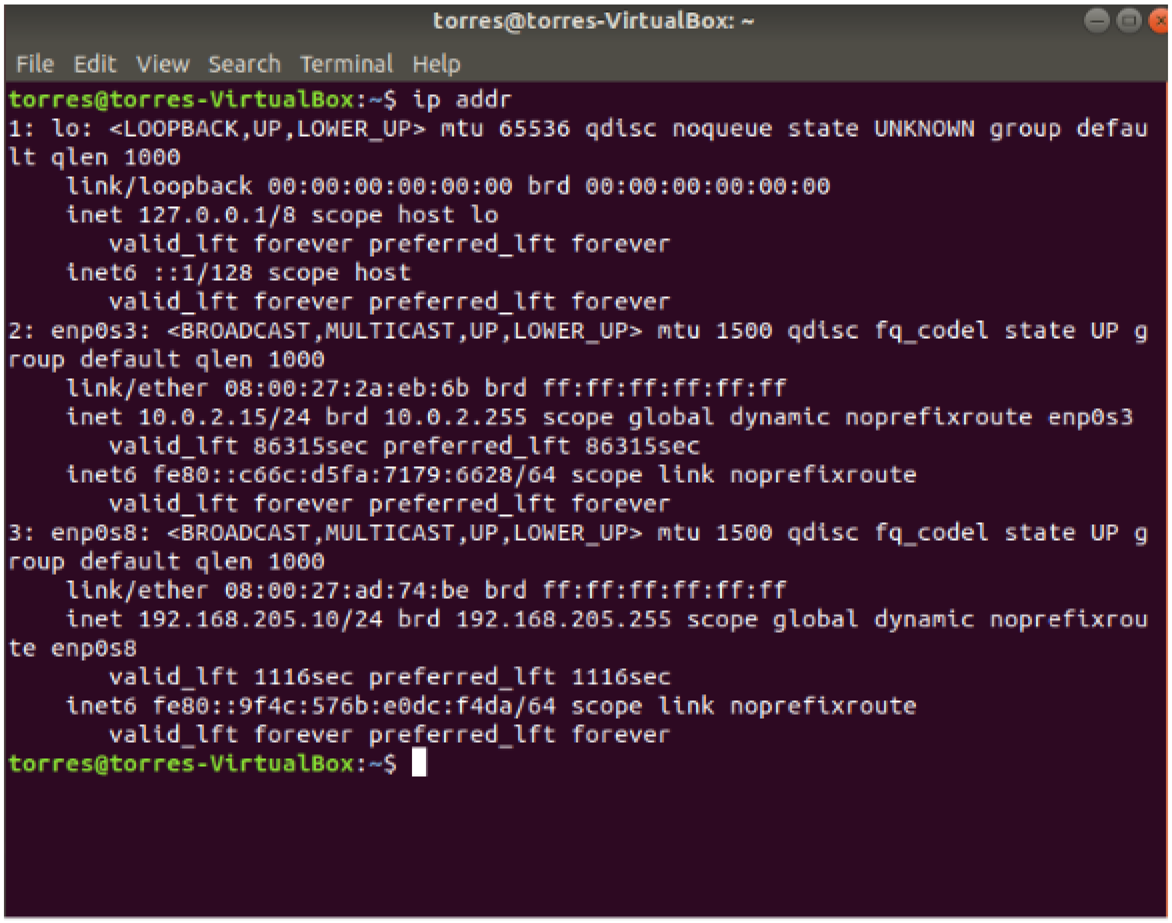

torres@torres-VirtualBox: ~
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                    /hostname                     Modified

pd-slave2




                                  [ New File ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

4th Step:

Now let’s figure out what our ip address is. To do that just type the command:

```
ip addr
```


```
torres@torres-VirtualBox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:2a:eb:6b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86315sec preferred_lft 86315sec
    inet6 fe80::c66c:d5fa:7179:6628/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:ad:74:be brd ff:ff:ff:ff:ff:ff
    inet 192.168.205.10/24 brd 192.168.205.255 scope global dynamic noprefixrou
te enp0s8
       valid_lft 1116sec preferred_lft 1116sec
    inet6 fe80::9f4c:576b:e0dc:f4da/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
torres@torres-VirtualBox:~$ 
```

This is on the master VM, as you can see our IP is 192.168.205.10. For you this will be different.

This means that our IP's are:

master: 192.168.205.10

slave1: 192.168.205.11

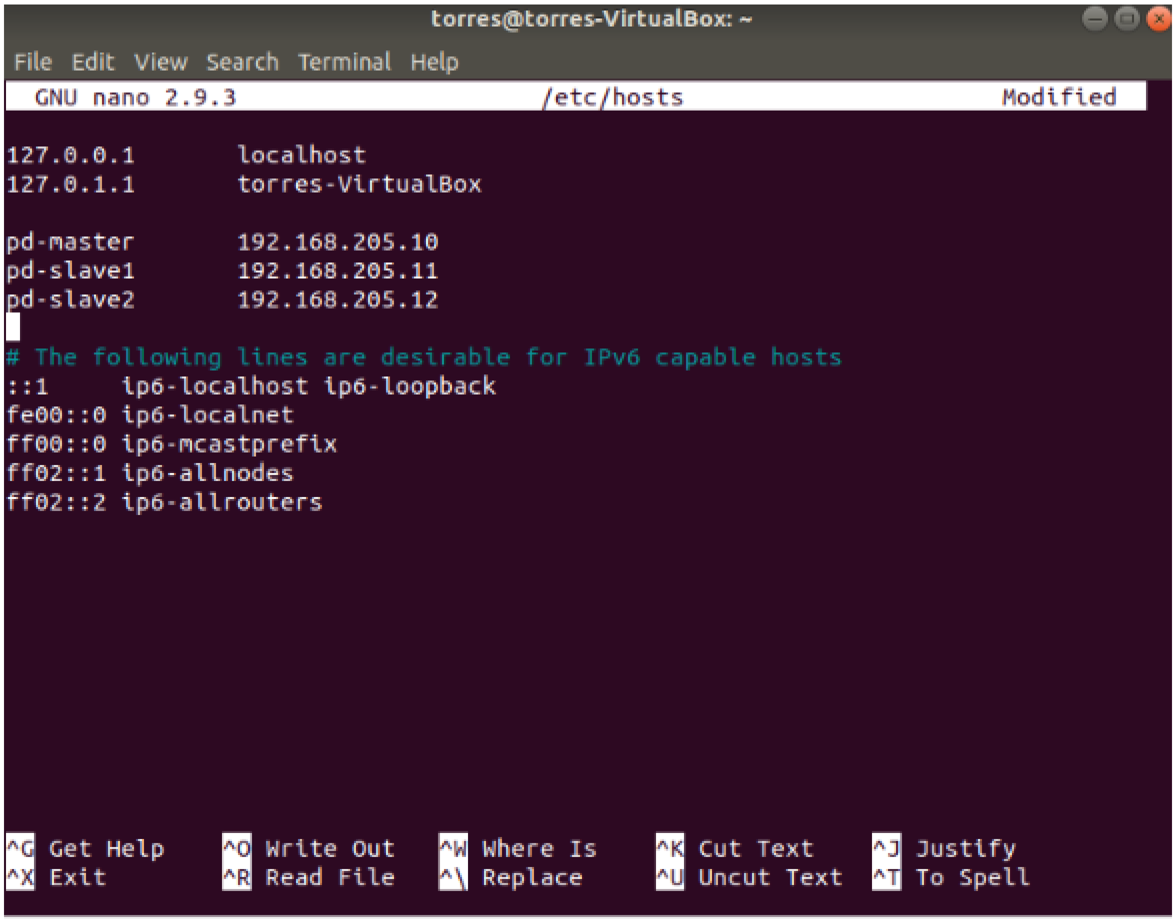
slave2: 192.168.205.12

5th Step:

We need to edit the **hosts** file. Use the following command:

```
sudo nano /etc/hosts
```

and add your network information:


torres@torres-VirtualBox: ~
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                    /etc/hosts                        Modified

127.0.0.1       localhost
127.0.1.1       torres-VirtualBox

pd-master       192.168.205.10
pd-slave1       192.168.205.11
pd-slave2       192.168.205.12

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

6th Step:

In order for the machines to assimilate the previous steps we need to reboot them. Use the following command in all of them:

```
sudo reboot
```

7th Step:

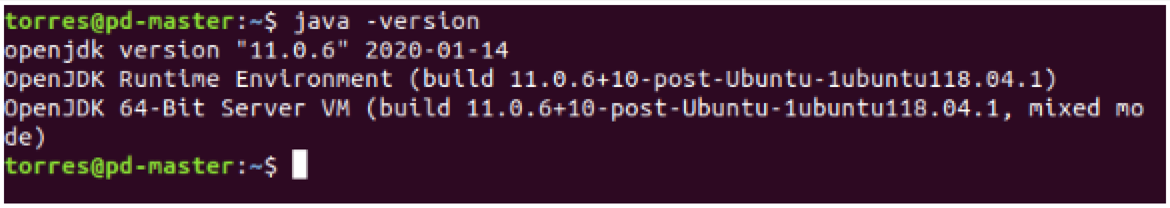
Do this step on all the Machines, master and slaves.

Now, in order to install Java we need to do some things. Follow these commands and give permission when needed:

```
$ sudo apt-get install software_properties_common
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install openjdk-11-jdk
```

To check if java is installed, run the following command.

```
$ java -version
```

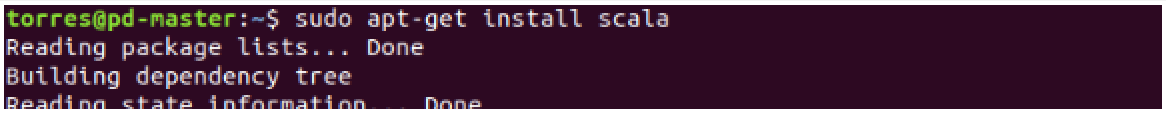


```
torres@pd-master:~$ java -version
openjdk version "11.0.6" 2020-01-14
OpenJDK Runtime Environment (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1)
OpenJDK 64-Bit Server VM (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1, mixed mo
de)
torres@pd-master:~$
```

8th Step:

Now let's install Scala on the master and the slaves. Use this command:

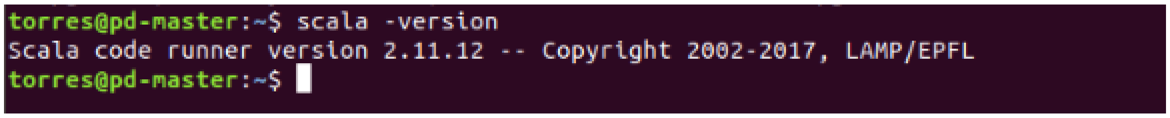
```
$ sudo apt-get install scala
```



```
torres@pd-master:~$ sudo apt-get install scala
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

To check if Scala was correctly installed run this command:

```
$ scala -version
```



```
torres@pd-master:~$ scala -version
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
torres@pd-master:~$
```

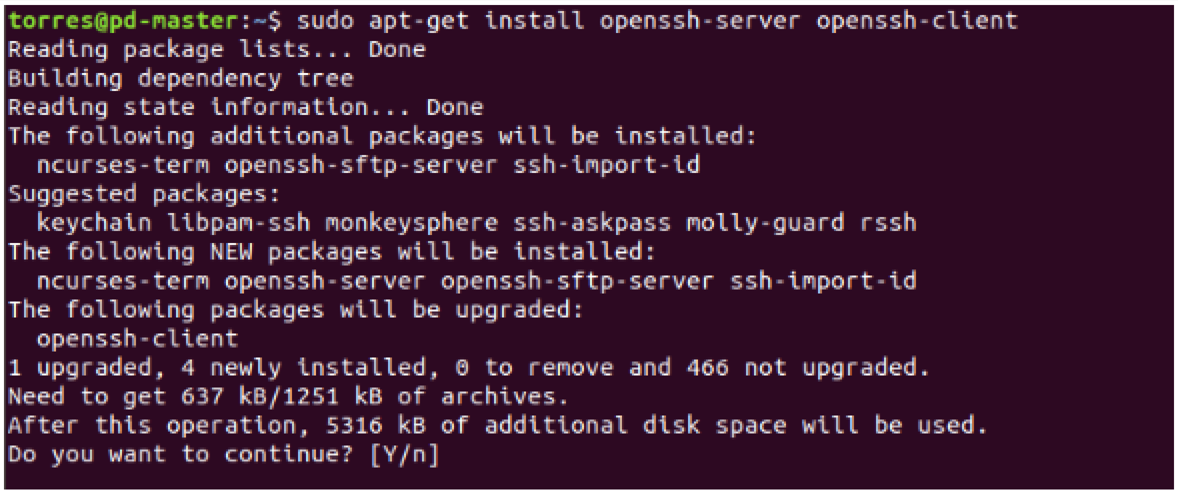
As you can see, Scala version 2.11.12 is now installed on my machine.

9th Step:

We will configure SSH, but this step in on master only.

We need to install the **Open SSH Server-Client**, use the command:

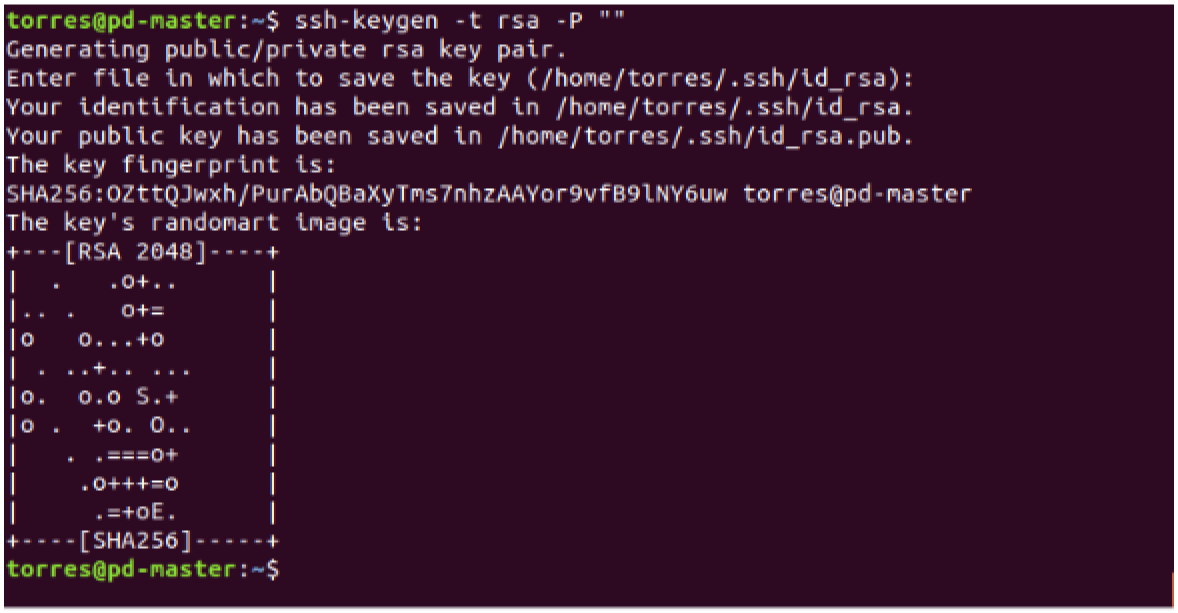
```
$ sudo apt-get install openssh-server openssh-client
```



```
torres@pd-master:~$ sudo apt-get install openssh-server openssh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard rssh
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 466 not upgraded.
Need to get 637 kB/1251 kB of archives.
After this operation, 5316 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Now generate key pairs:

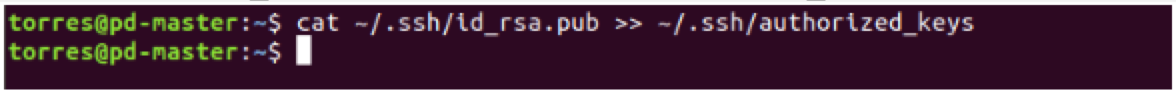
```
$ ssh-keygen -t rsa -P ""
```



```
torres@pd-master:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/torres/.ssh/id_rsa):
Your identification has been saved in /home/torres/.ssh/id_rsa.
Your public key has been saved in /home/torres/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:OZttQJwxh/PurAbQBaXyTms7nhzAAYor9vfB9lNY6uw torres@pd-master
The key's randomart image is:
+---[RSA 2048]----+
|  .   .o+..      |
|.. .   o+=       |
|o   o...+o       |
| . ..+.. ...     |
|o.  o.o S.+      |
|o .  +o. O..     |
|   . .===o+      |
|    .o+++=o      |
|     .=+oE.      |
+----[SHA256]-----+
torres@pd-master:~$
```

Use the following command in order to make this key an authorized one:

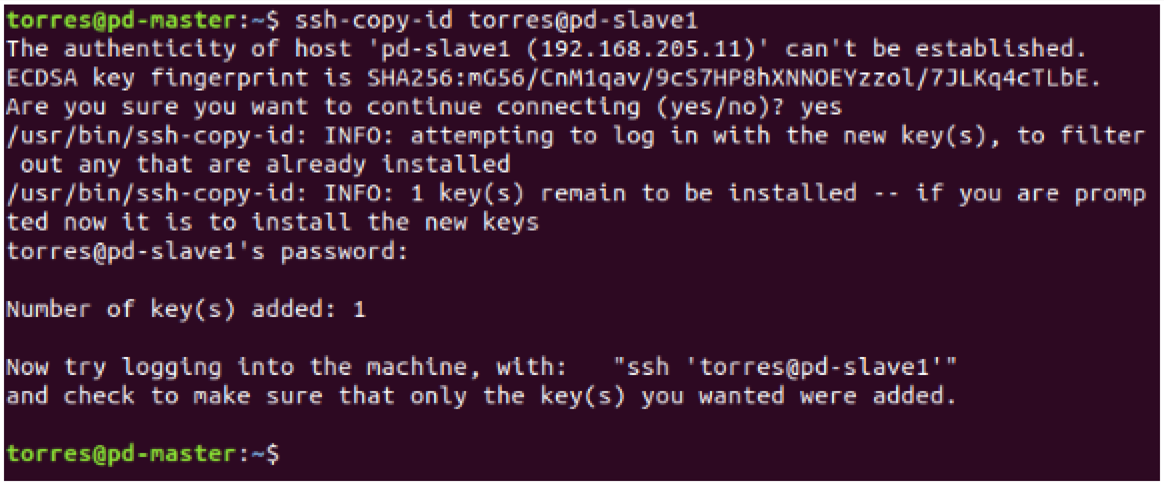
```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```



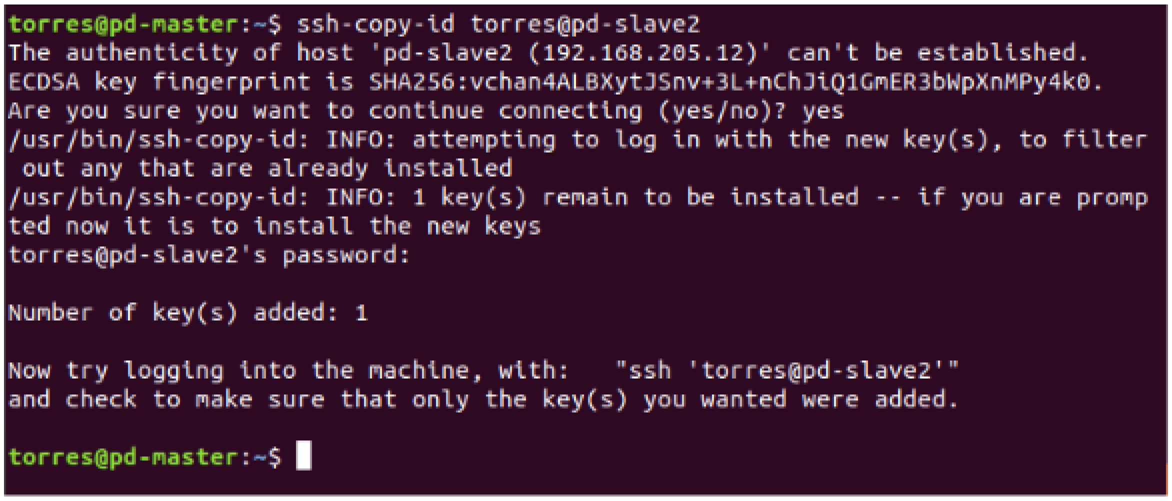
```
torres@pd-master:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
torres@pd-master:~$ 
```

Now we need to copy the content of *.ssh/id_rsa.pub* (of master) to *.ssh/authorized_keys* (of all the slaves as well as master). Use these commands:

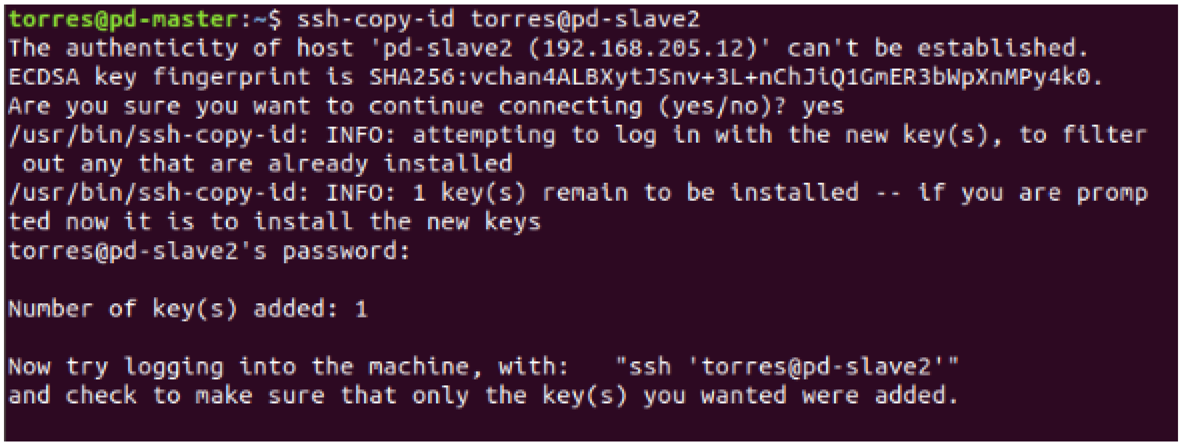
```
ssh-copy-id user@pd-master
ssh-copy-id user@pd-slave1
ssh-copy-id user@pd-slave2
```



```
torres@pd-master:~$ ssh-copy-id torres@pd-slave1
The authenticity of host 'pd-slave1 (192.168.205.11)' can't be established.
ECDSA key fingerprint is SHA256:mG56/CnM1qav/9cS7HP8hXNNOEYzzol/7JLKq4cTLbE.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
torres@pd-slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'torres@pd-slave1'"
and check to make sure that only the key(s) you wanted were added.

torres@pd-master:~$
```



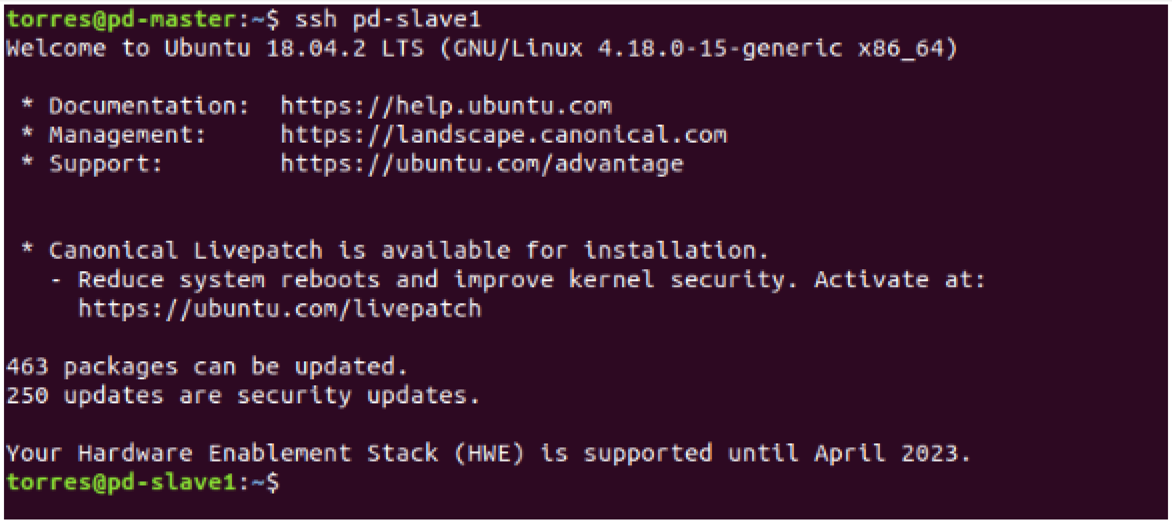
```
torres@pd-master:~$ ssh-copy-id torres@pd-slave2
The authenticity of host 'pd-slave2 (192.168.205.12)' can't be established.
ECDSA key fingerprint is SHA256:vchan4ALBXytJSnv+3L+nChJiQ1GmER3bWpXnMPy4k0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
torres@pd-slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'torres@pd-slave2'"
and check to make sure that only the key(s) you wanted were added.

torres@pd-master:~$ 
```



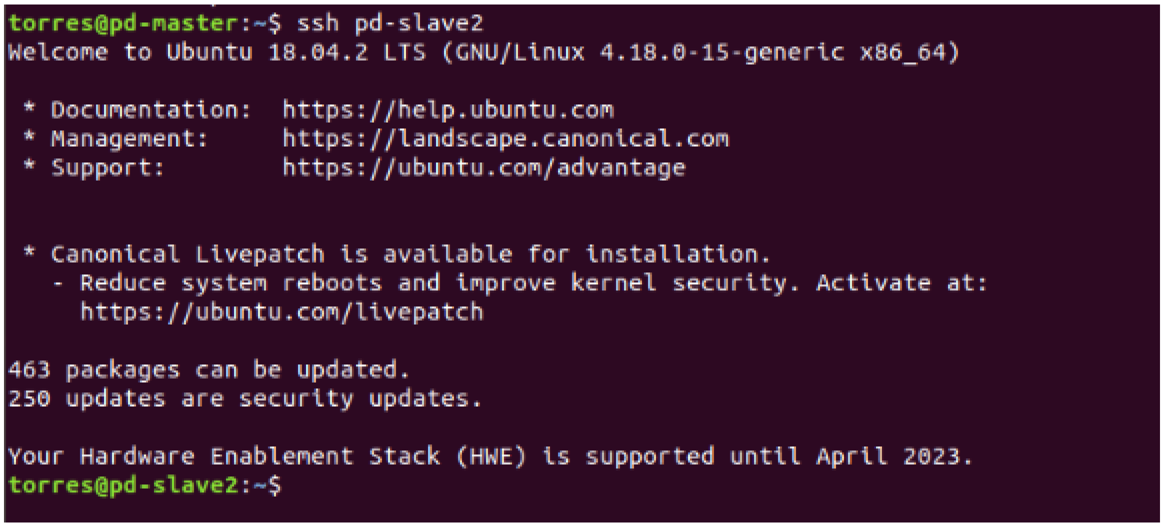
```
torres@pd-master:~$ ssh-copy-id torres@pd-slave2
The authenticity of host 'pd-slave2 (192.168.205.12)' can't be established.
ECDSA key fingerprint is SHA256:vchan4ALBXytJSnv+3L+nChJiQ1GmER3bWpXnMPy4k0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
torres@pd-slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'torres@pd-slave2'"
and check to make sure that only the key(s) you wanted were added.
```

Let's check if everything went well, try to connect to the slaves:

```
$ ssh slave01
$ ssh slave02
```



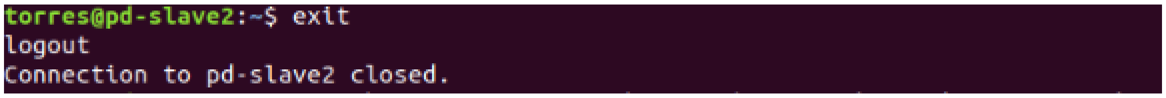
```
torres@pd-master:~$ ssh pd-slave1
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

463 packages can be updated.
250 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
torres@pd-slave1:~$
```



```
torres@pd-master:~$ ssh pd-slave2
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

463 packages can be updated.
250 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
torres@pd-slave2:~$
```

As you can see everything went well, to exit just type the command:

```
exit
```



```
torres@pd-slave2:~$ exit
logout
Connection to pd-slave2 closed.
```

10th Step:

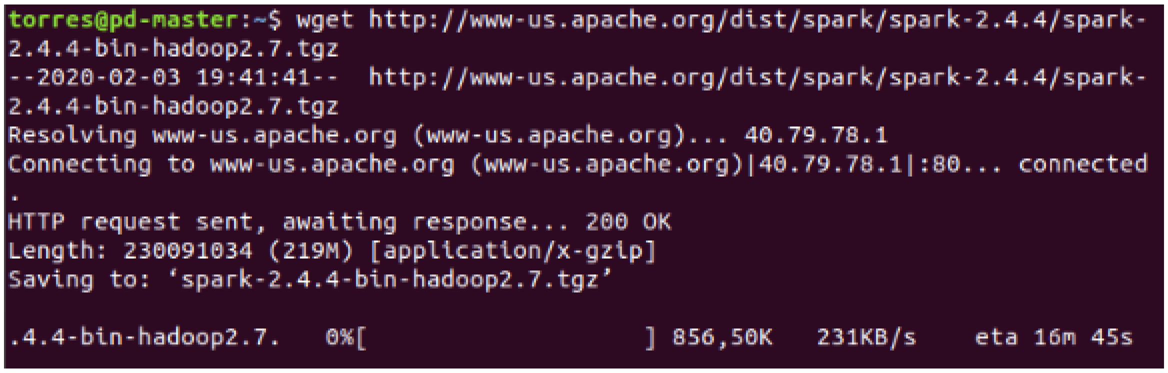
Now we Download the latest version of Apache Spark.

NOTE: Everything inside this step must be done on all the virtual machines.

Use the following command :

```
$ wget
http://www-us.apache.org/dist/spark/spark-2.4.4/spark-2.4.4-bin-hadoop2.7.tgz
```

This is the most recent version as of the writing of this arcticle, it might have changed if you try it later. Anyway, I think you'll still be good using this one.

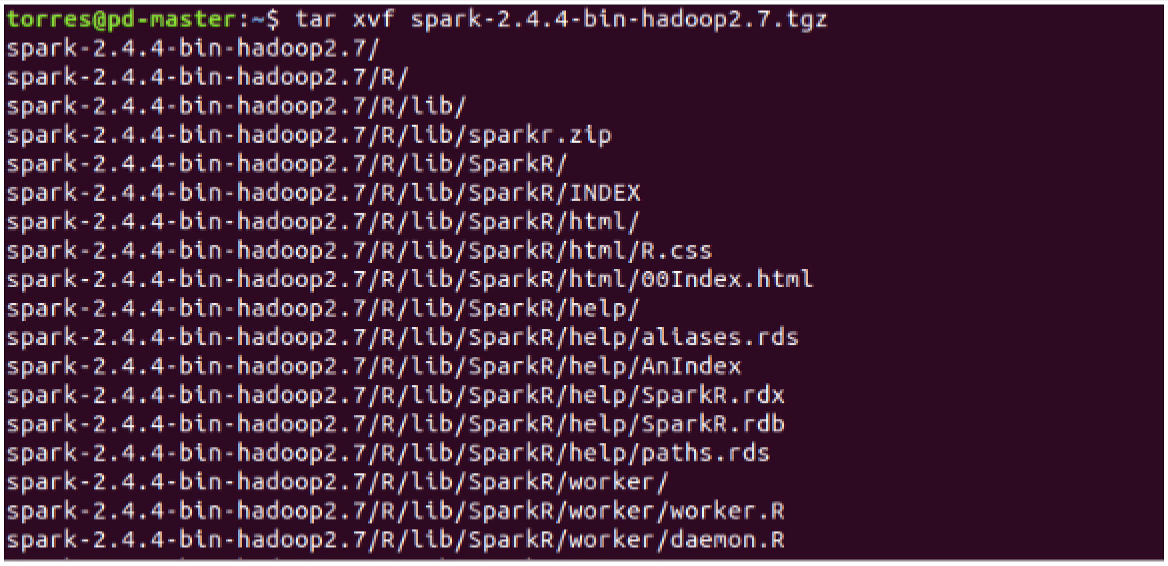


```
torres@pd-master:~$ wget http://www-us.apache.org/dist/spark/spark-2.4.4/spark-
2.4.4-bin-hadoop2.7.tgz
--2020-02-03 19:41:41--  http://www-us.apache.org/dist/spark/spark-2.4.4/spark-
2.4.4-bin-hadoop2.7.tgz
Resolving www-us.apache.org (www-us.apache.org)... 40.79.78.1
Connecting to www-us.apache.org (www-us.apache.org)|40.79.78.1|:80... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 230091034 (219M) [application/x-gzip]
Saving to: ‘spark-2.4.4-bin-hadoop2.7.tgz’

.4.4-bin-hadoop2.7.   0%[                    ] 856,50K   231KB/s    eta 16m 45s
```

Extract the Apache Spark file you just downloaded

Use the following command to extract the Spark tar file:

```
$ tar xvf spark-2.4.4-bin-hadoop2.7.tgz
```

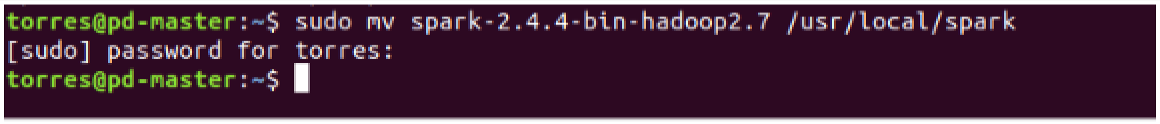


```
torres@pd-master:~$ tar xvf spark-2.4.4-bin-hadoop2.7.tgz
spark-2.4.4-bin-hadoop2.7/
spark-2.4.4-bin-hadoop2.7/R/
spark-2.4.4-bin-hadoop2.7/R/lib/
spark-2.4.4-bin-hadoop2.7/R/lib/sparkr.zip
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/INDEX
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/html/
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/html/R.css
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/html/00Index.html
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/aliases.rds
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/AnIndex
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/SparkR.rdx
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/SparkR.rdb
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/help/paths.rds
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/worker/
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/worker/worker.R
spark-2.4.4-bin-hadoop2.7/R/lib/SparkR/worker/daemon.R
```

Move Apache Spark software files

Use the following command to move the spark software files to respective directory (*/usr/local/bin*)

```
$ sudo mv spark-2.4.4-bin-hadoop2.7 /usr/local/spark
```

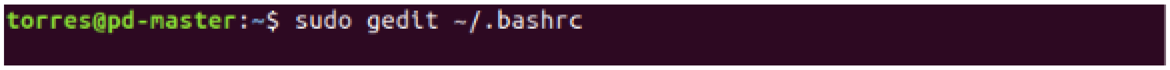


```
torres@pd-master:~$ sudo mv spark-2.4.4-bin-hadoop2.7 /usr/local/spark
[sudo] password for torres:
torres@pd-master:~$
```

Set up the environment for Apache Spark

Edit the *bashrc* file using this command:

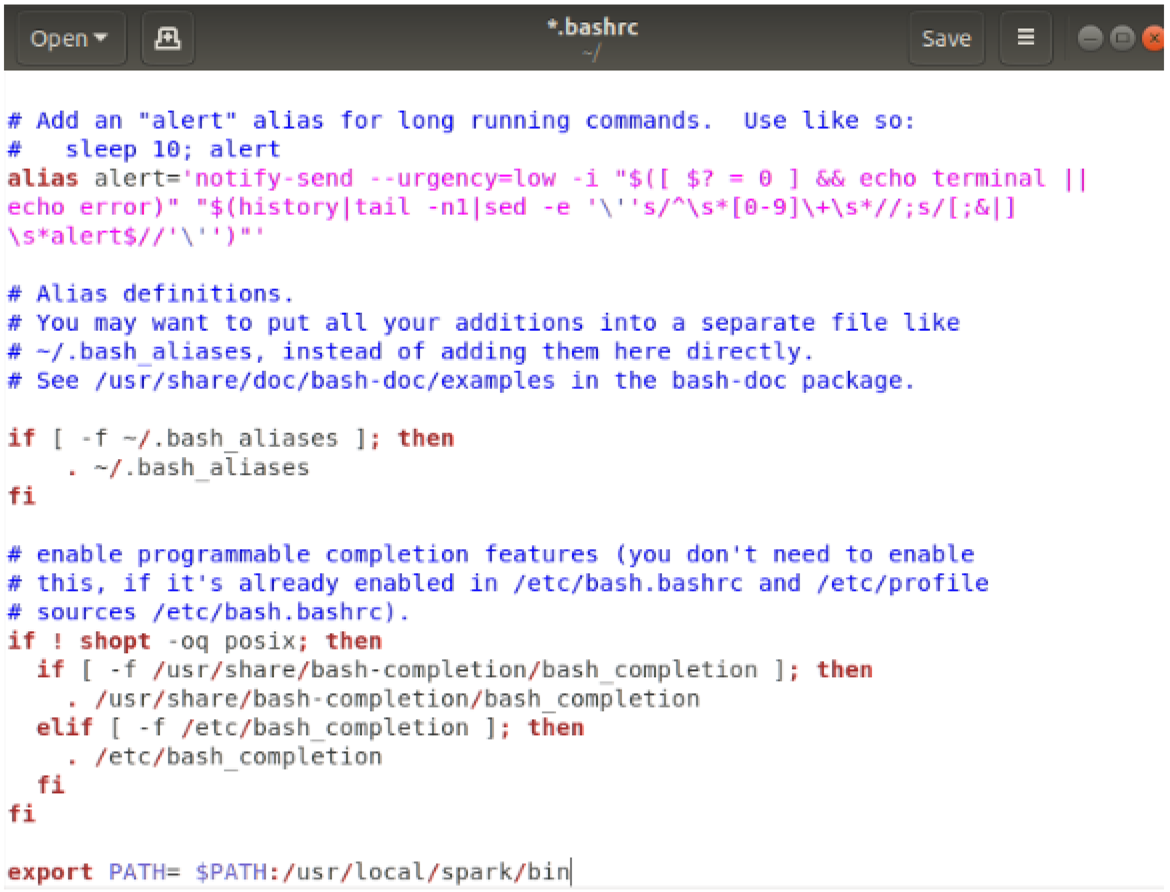
```
$ sudo gedit~/.bashrc
```



```
torres@pd-master:~$ sudo gedit ~/.bashrc
```

Add the following line to the file. This adds the location where the spark software file are located to the PATH variable.

```
export PATH = $PATH:/usr/local/spark/bin
```

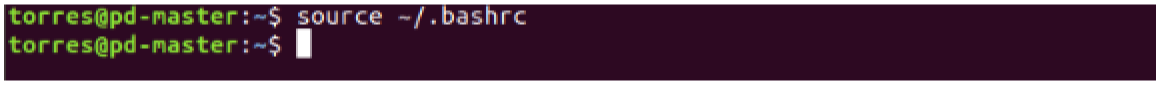


```
# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal ||
echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]
\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH= $PATH:/usr/local/spark/bin
```

Note: this screenshot has a mistake, when you're doing this don't leave a space like I did. Just write "PATH=$PATH".

Now we need to use the following command for sourcing the *~/.bashrc* file:

```
$ source ~/.bashrc
```



```
torres@pd-master:~$ source ~/.bashrc
torres@pd-master:~$ 
```

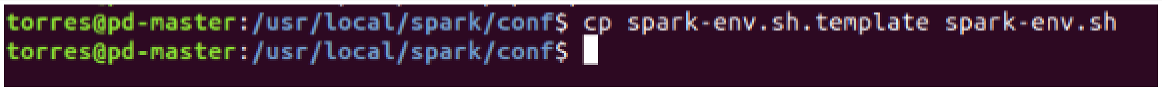
11th Step:

Apache Spark Master Configuration (do this step on the Master VM only)

Edit spark-env.sh

Move to spark *conf* folder and create a copy of the template of *spark-env.sh* and rename it.

```
$ cd /usr/local/spark/conf
$ cp spark-env.sh.template spark-env.sh
```



```
torres@pd-master:/usr/local/spark/conf$ cp spark-env.sh.template spark-env.sh
torres@pd-master:/usr/local/spark/conf$ 
```

Now edit the configuration file *spark-env.sh*.

```
$ sudo vim spark-env.sh
```

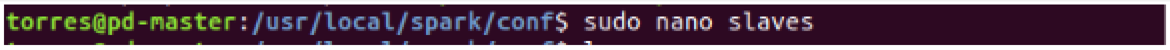
And add the following parameters:

```
export SPARK_MASTER_HOST='<MASTER-IP>'export
JAVA_HOME=<Path_of_JAVA_installation>
```

Add Workers

Edit the configuration file *slaves* in (*/usr/local/spark/con*f).

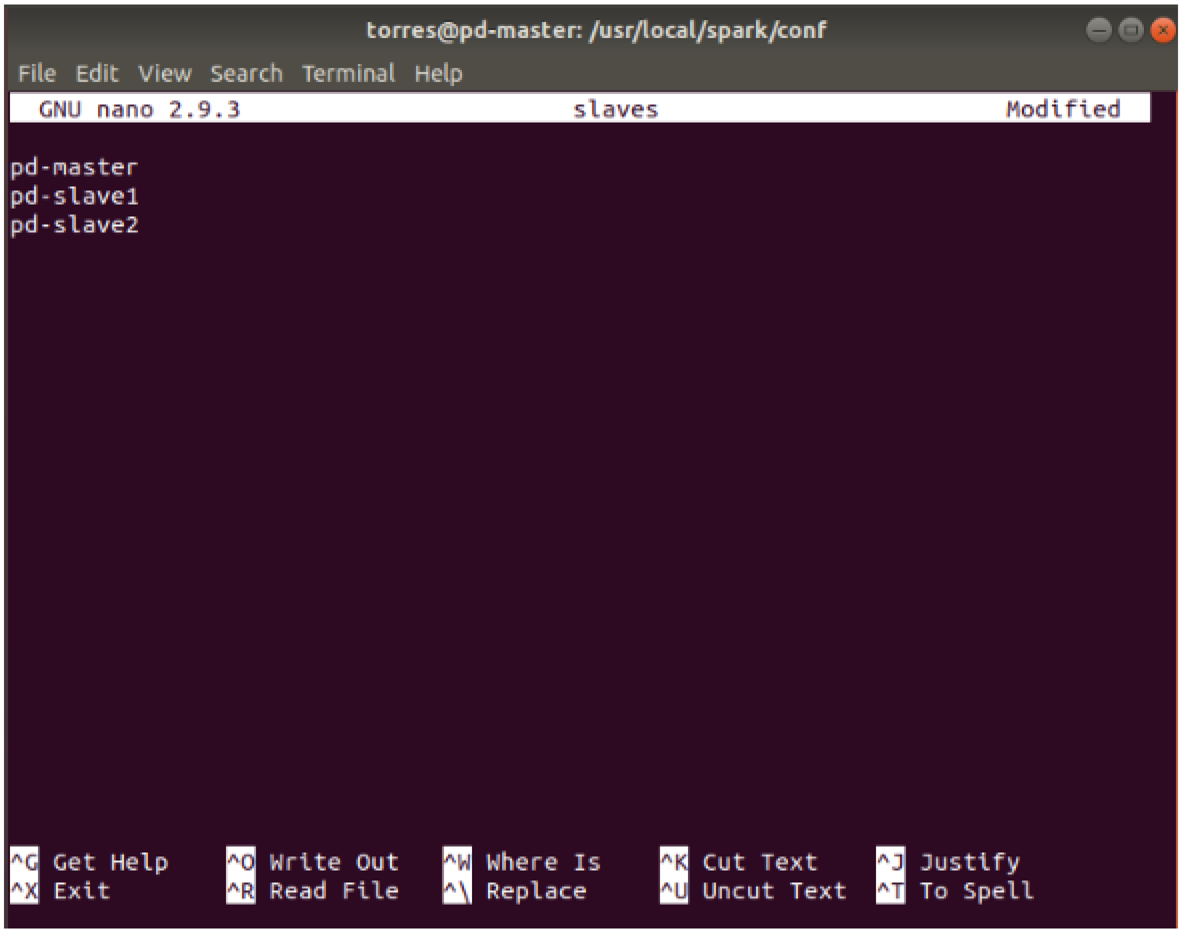
```
$ sudo nano slaves
```



```
torres@pd-master:/usr/local/spark/conf$ sudo nano slaves
```

And add the following entries.

```
pd-master
pd-slave01
pd-slave02
```


torres@pd-master: /usr/local/spark/conf
File Edit View Search Terminal Help

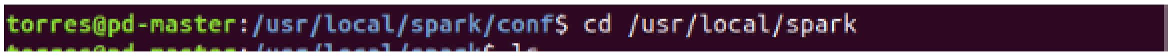
```
  GNU nano 2.9.3                    slaves                          Modified

pd-master
pd-slave1
pd-slave2
```

12th Step:

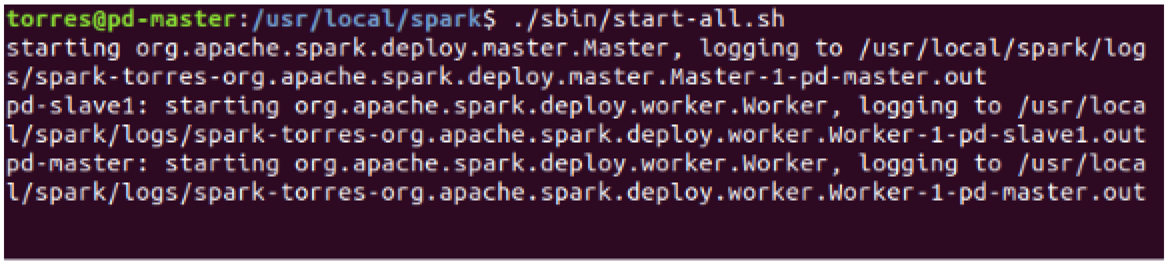
Let’s try to start our Apache Spark Cluster, hopefully everything is ok!

To start the spark cluster, run the following command on master.:

```
$ cd /usr/local/spark
$ ./sbin/start-all.sh
```



```
torres@pd-master:/usr/local/spark/conf$ cd /usr/local/spark
```



```
torres@pd-master:/usr/local/spark$ ./sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark/log
s/spark-torres-org.apache.spark.deploy.master.Master-1-pd-master.out
pd-slave1: starting org.apache.spark.deploy.worker.Worker, logging to /usr/loca
l/spark/logs/spark-torres-org.apache.spark.deploy.worker.Worker-1-pd-slave1.out
pd-master: starting org.apache.spark.deploy.worker.Worker, logging to /usr/loca
l/spark/logs/spark-torres-org.apache.spark.deploy.worker.Worker-1-pd-master.out
```

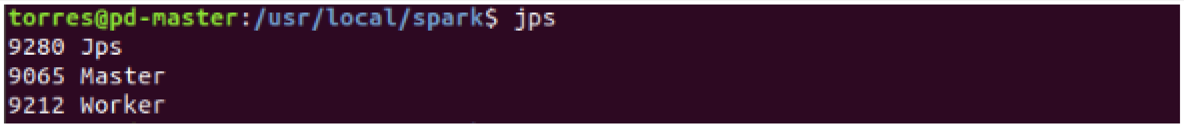
I won't stop it, but in case you want to stop the cluster, this is the command:

```
$ ./sbin/stop-all.sh
```

13th Step:

To check if the services started we use the command:

```
$ jps
```

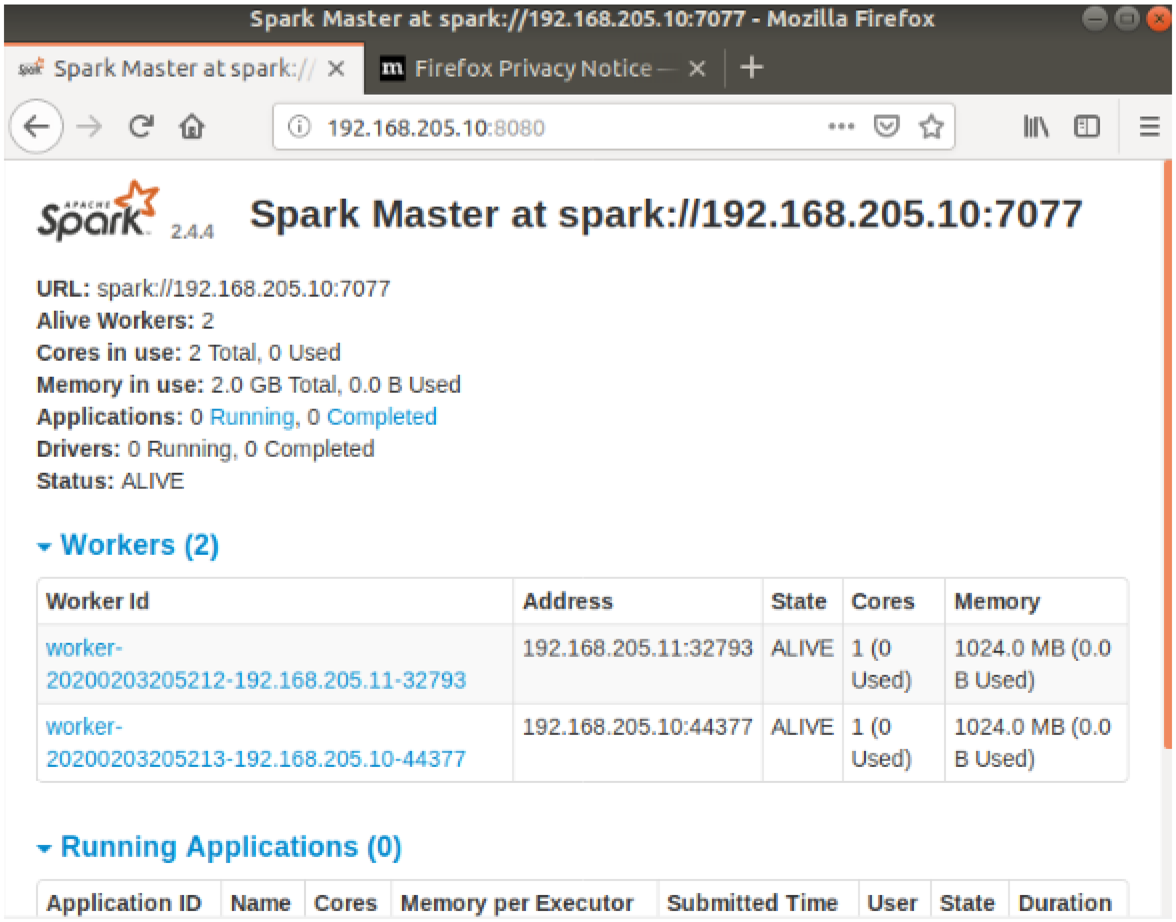


```
torres@pd-master:/usr/local/spark$ jps
9280 Jps
9065 Master
9212 Worker
```

14th Step:

Browse the Spark UI to know about your cluster. To do this, go to your browser and type:

```
http://<MASTER-IP>:8080/
```



Spark Master at spark://192.168.205.10:7077

URL: spark://192.168.205.10:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20200203205212-192.168.205.11-32793	192.168.205.11:32793	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20200203205213-192.168.205.10-44377	192.168.205.10:44377	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

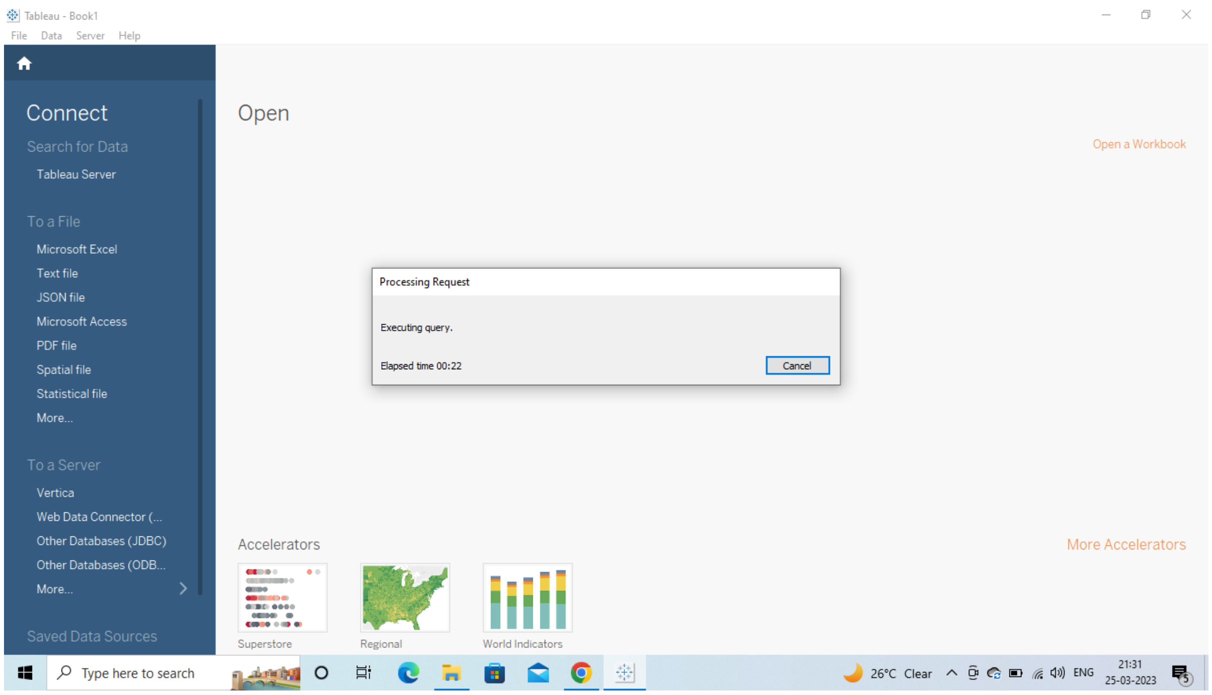
As you can see we have 2 Alive Workers, our slaves, which means it’s all done!

Assignment 8

Title: To Perform Graph analytics and visualization using Tableau.
SOFTWARE REQUIREMENTS:
1.windows computer or Mac computer
2. Tableau

THEORY:
Tableau is a data analytics and visualization tool used widely in the industry today. Many businesses even consider it indispensable for data-science-related work. Tableau's ease of use comes from the fact that it has a drag and drop interface. This feature helps to perform tasks like sorting, comparing and analyzing, very easily and fast. Tableau is also compatible with multiple sources, including Excel, SQL Server, and cloud-based data repositories which makes it an excellent choice for Data Scientists.

Step1: Download and install the Tableau.



Processing Request

Executing query.

Elapsed time 00:22

image 1: Tableau

Tableau Workspace

The Tableau workspace is a collection of worksheets, menu bar, toolbar, marks card, shelves and a lot of other elements. Sheets can be worksheets, dashboards, or stories. The image below highlights the major components of the workspace. However, more familiarity will be achieved once we work with actual data.

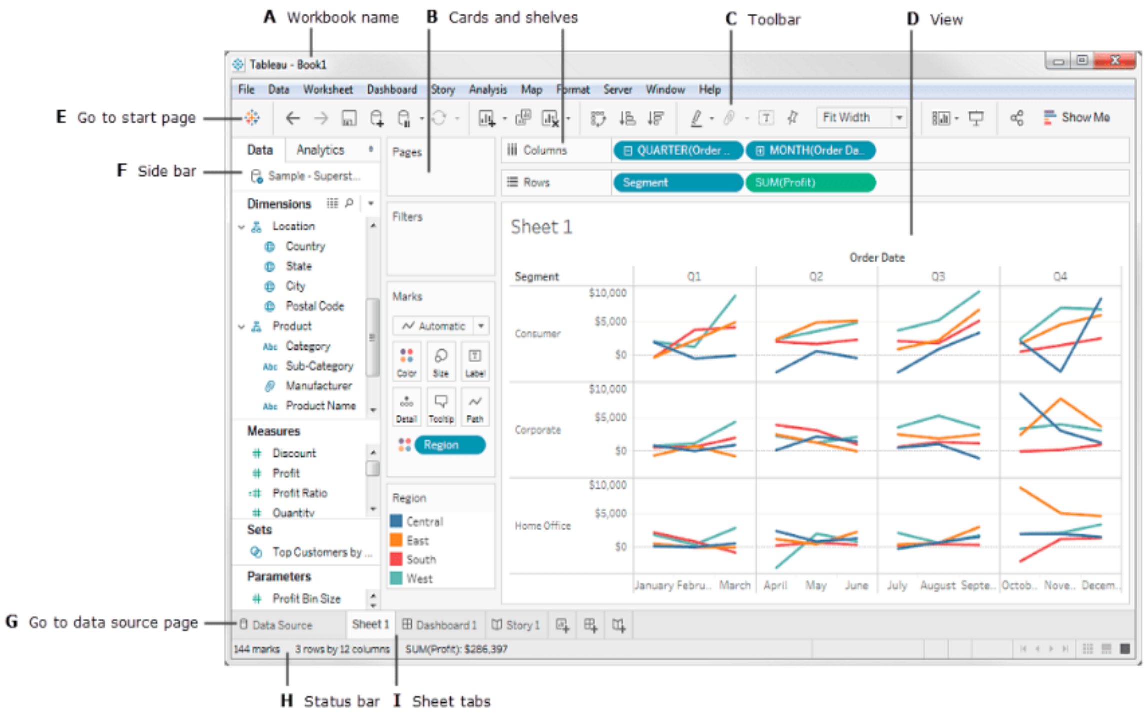


Image 2: Tablue Workspace

Step2: Download the sample data file in excel.

Tableau is compatible with a lot of data sources. The data sources supported by Tableau appear on the left side of the opening screen. Some commonly used data sources are excel, text file, relational database or even on a server.

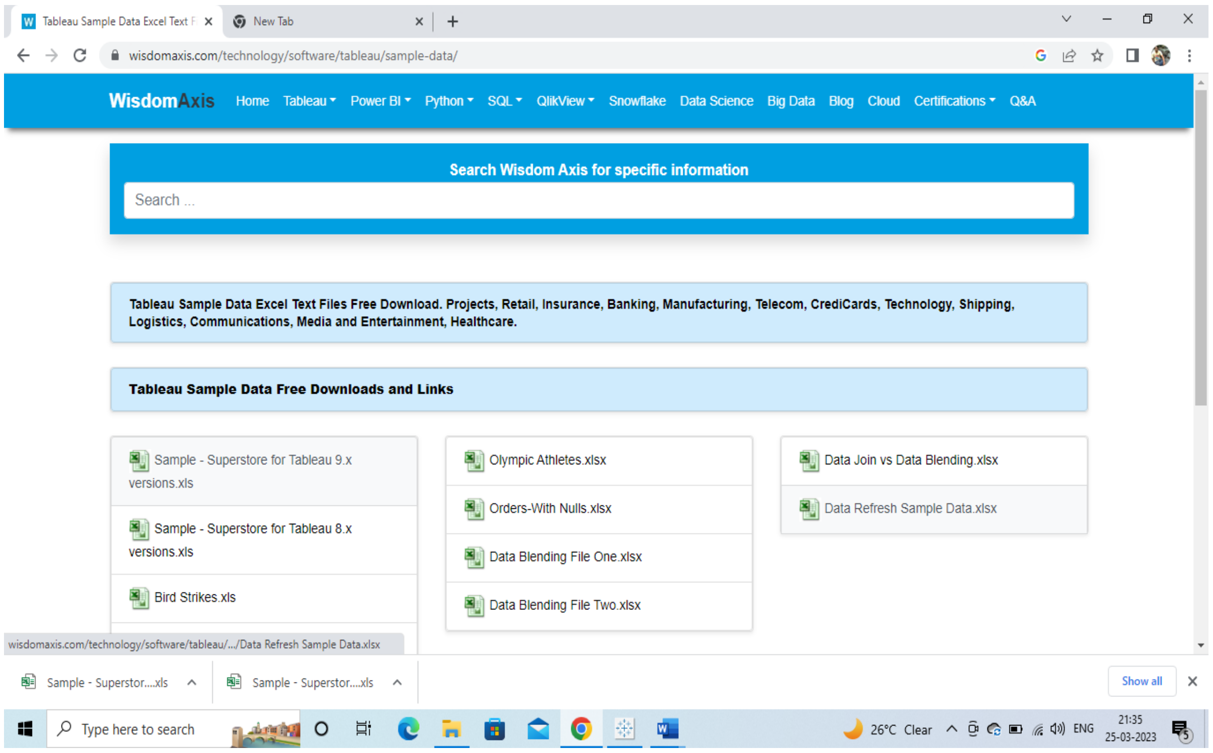


Tableau Sample Data Excel Text F

New Tab

wisdomaxis.com/technology/software/tableau/sample-data/

Search Wisdom Axis for specific information

Search ...

Tableau Sample Data Excel Text Files Free Download. Projects, Retail, Insurance, Banking, Manufacturing, Telecom, CrediCards, Technology, Shipping, Logistics, Communications, Media and Entertainment, Healthcare.

Tableau Sample Data Free Downloads and Links

Sample - Superstore for Tableau 9.x versions.xls	Olympic Athletes.xlsx	Data Join vs Data Blending.xlsx
Sample - Superstore for Tableau 8.x versions.xls	Orders-With Nulls.xlsx	Data Refresh Sample Data.xlsx
Bird Strikes.xls	Data Blending File One.xlsx	
	Data Blending File Two.xlsx	

wisdomaxis.com/technology/software/tableau/.../Data Refresh Sample Data.xlsx

Sample - Superstor....xls

Sample - Superstor....xls

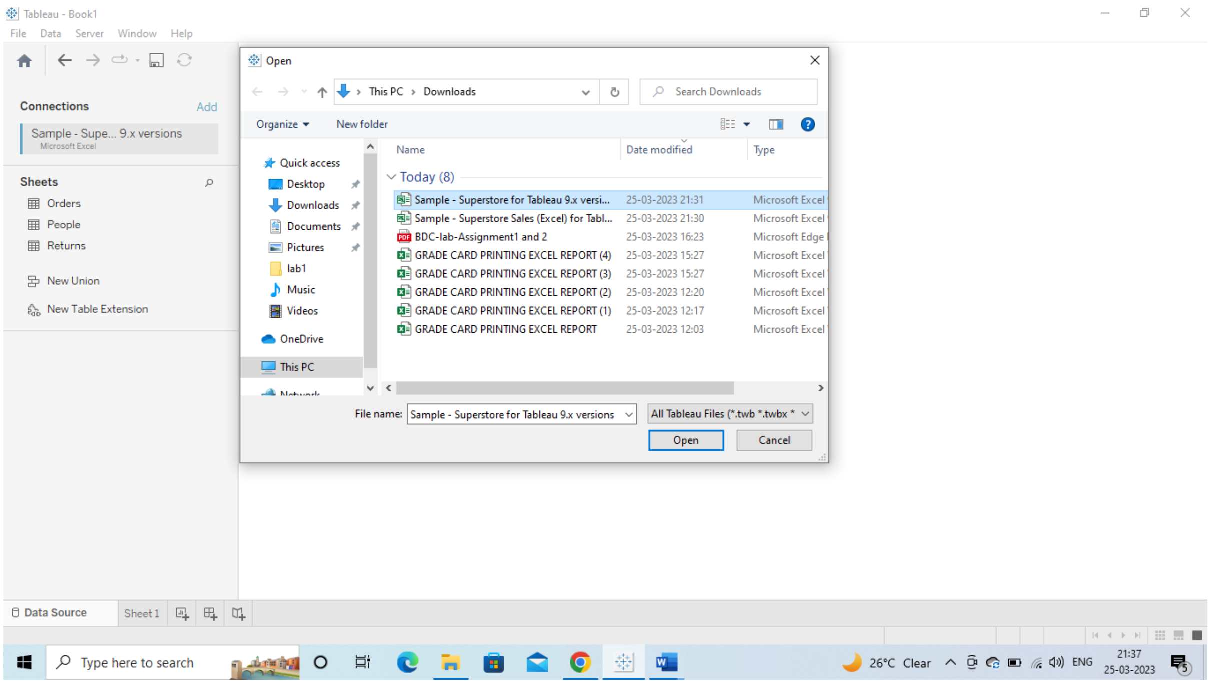
Show all

Type here to search

26°C Clear ENG 21:35 25-03-2023

Image 3: Sample Data for Tableau

Step 3: Select open menu open downloaded data file



Today (8)

Name	Date modified	Type
Sample - Superstore for Tableau 9.x versi...	25-03-2023 21:31	Microsoft Excel
Sample - Superstore Sales (Excel) for Tabl...	25-03-2023 21:30	Microsoft Excel
BDC-lab-Assignment1 and 2	25-03-2023 16:23	Microsoft Edge
GRADE CARD PRINTING EXCEL REPORT (4)	25-03-2023 15:27	Microsoft Excel
GRADE CARD PRINTING EXCEL REPORT (3)	25-03-2023 15:27	Microsoft Excel
GRADE CARD PRINTING EXCEL REPORT (2)	25-03-2023 12:20	Microsoft Excel
GRADE CARD PRINTING EXCEL REPORT (1)	25-03-2023 12:17	Microsoft Excel
GRADE CARD PRINTING EXCEL REPORT	25-03-2023 12:03	Microsoft Excel

File name: Sample - Superstore for Tableau 9.x versions

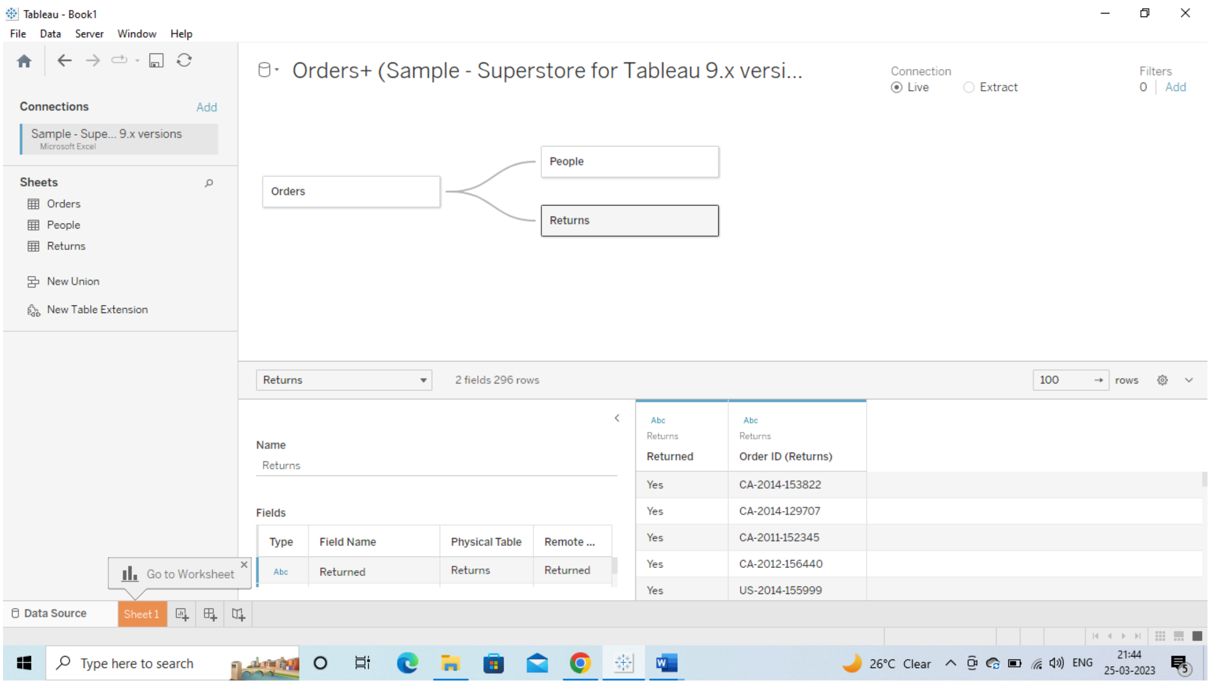
All Tableau Files (*.twb *.twbx *

Open Cancel

Image 4: Selection Of sample data file

Step 4: Create the connection between Data fields from the database.

Form sample data drag and drop the fileds. Here people, return and order select and drop on the workspace. Create the connection between them.



Orders+ (Sample - Superstore for Tableau 9.x versi...

Connection: (●) Live ( ) Extract

Filters: 0 Add

Returns — 2 fields 296 rows — 100 rows

Name: Returns

Fields

Type	Field Name	Physical Table	Remote ...
Abc	Returned	Returns	Returned

Returned	Order ID (Returns)
Yes	CA-2014-153822
Yes	CA-2014-129707
Yes	CA-2011-152345
Yes	CA-2012-156440
Yes	US-2014-155999

Image 5: Create Data field connection

Go to sheet1 for visualization of the data. Different visualization available in form of graph.

Step 5: Select row and column as customer and their respective country. Field selection done using available data fields.

Step 6: Select chart type for more visualization of the data.

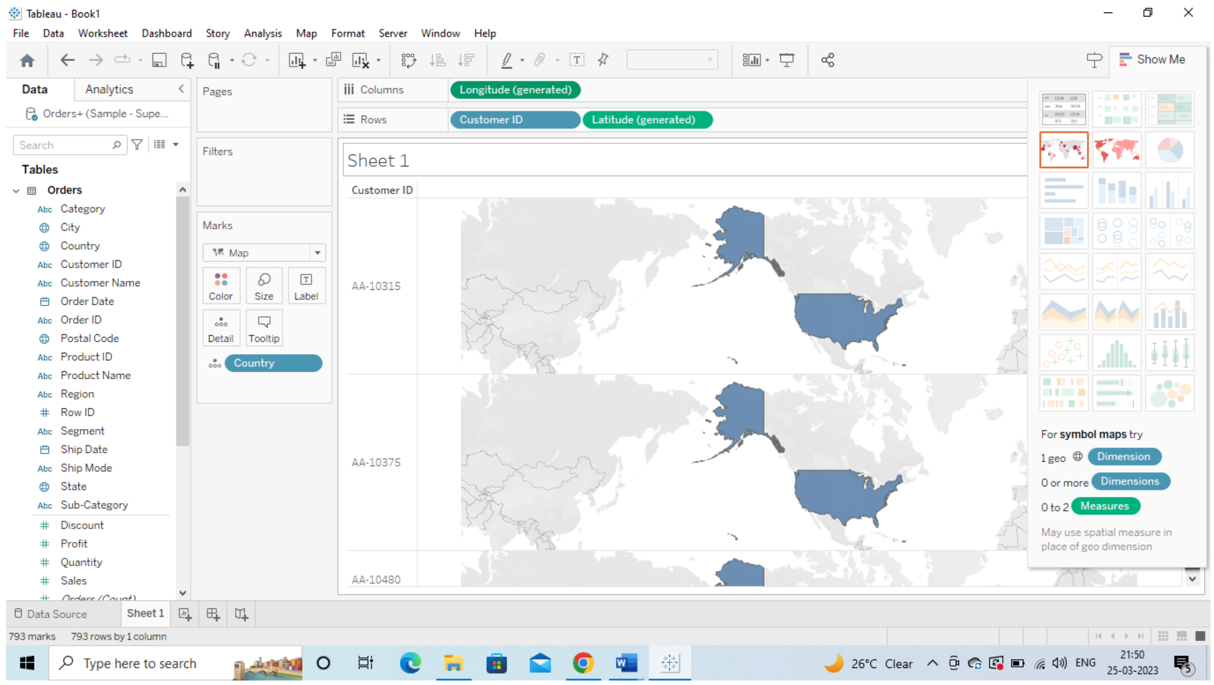
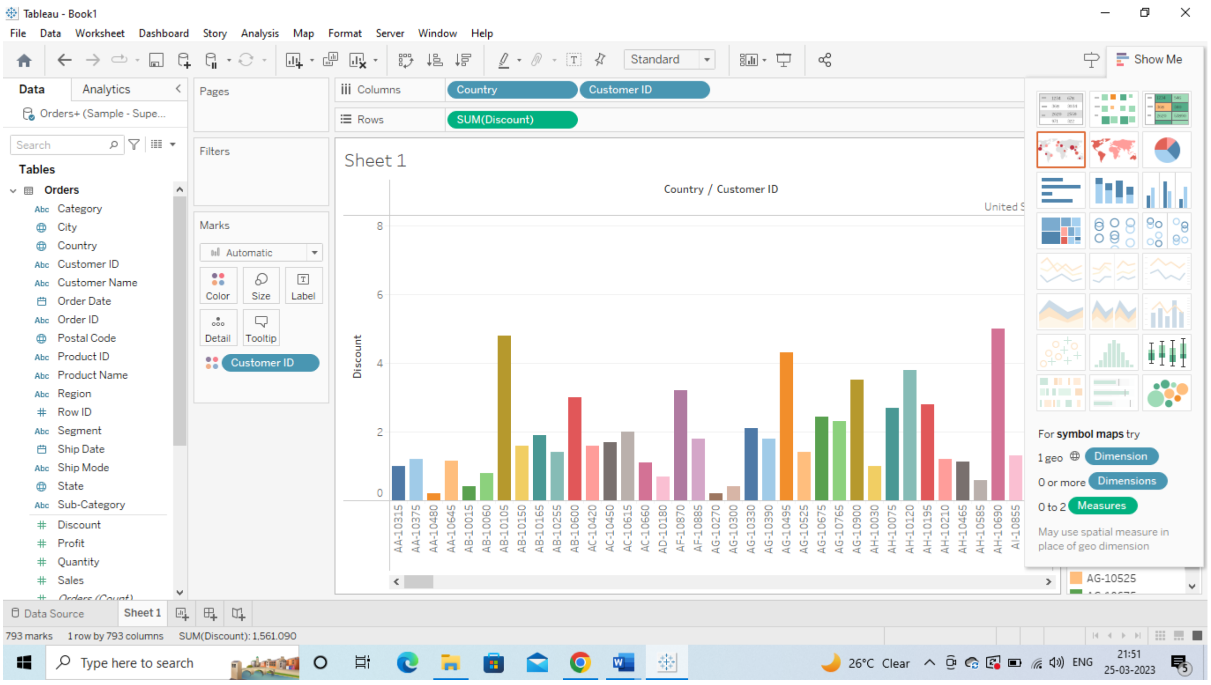


Image 6: Data visualization using map



Columns: Country, Customer ID

Rows: SUM(Discount)

Marks: Customer ID (Color)

793 marks 1 row by 793 columns SUM(Discount): 1,561.090

Image 7: Data visualization using Bar Chart

Conclusion: In such way Tableau Visualization Though the dataset is complex or the dataset is very big, in tableau, we can create dashboards very easily and within less time.