Practical 1

Aim: Hadoop Configuration and Single node cluster setup and perform file management task in Hadoop

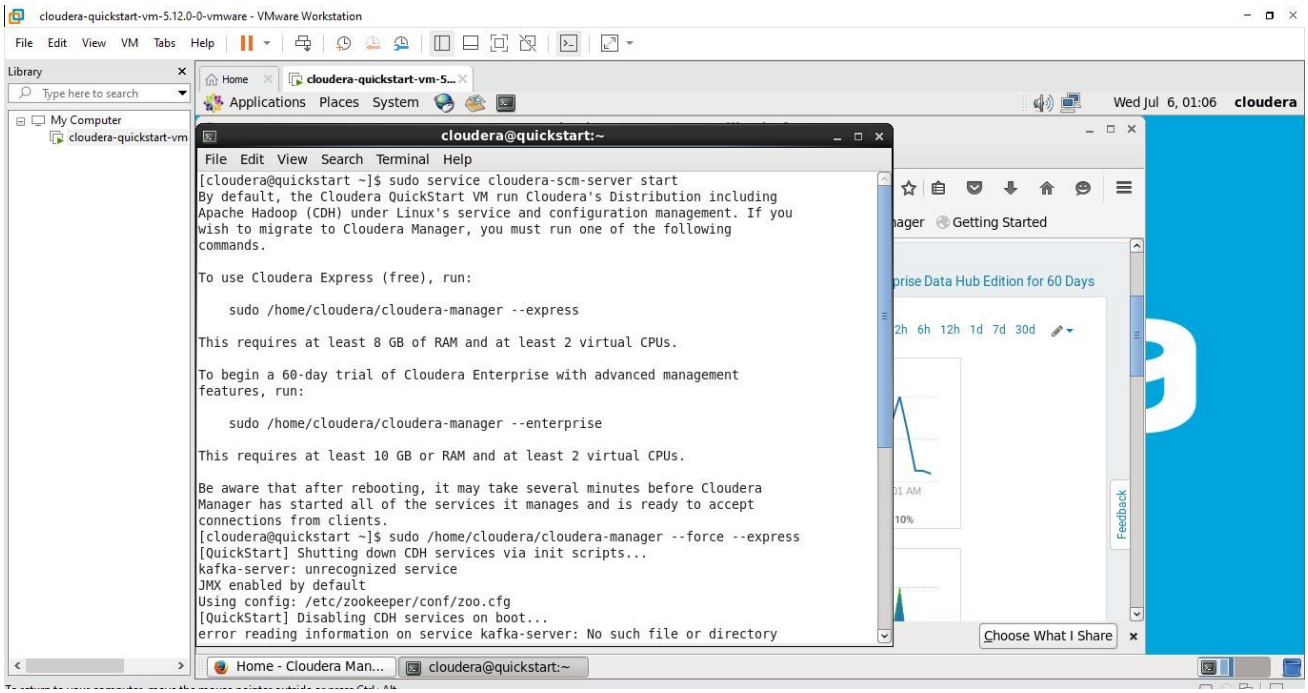
Aim: Hadoop Configuration and Single node cluster setup and perform file management task in

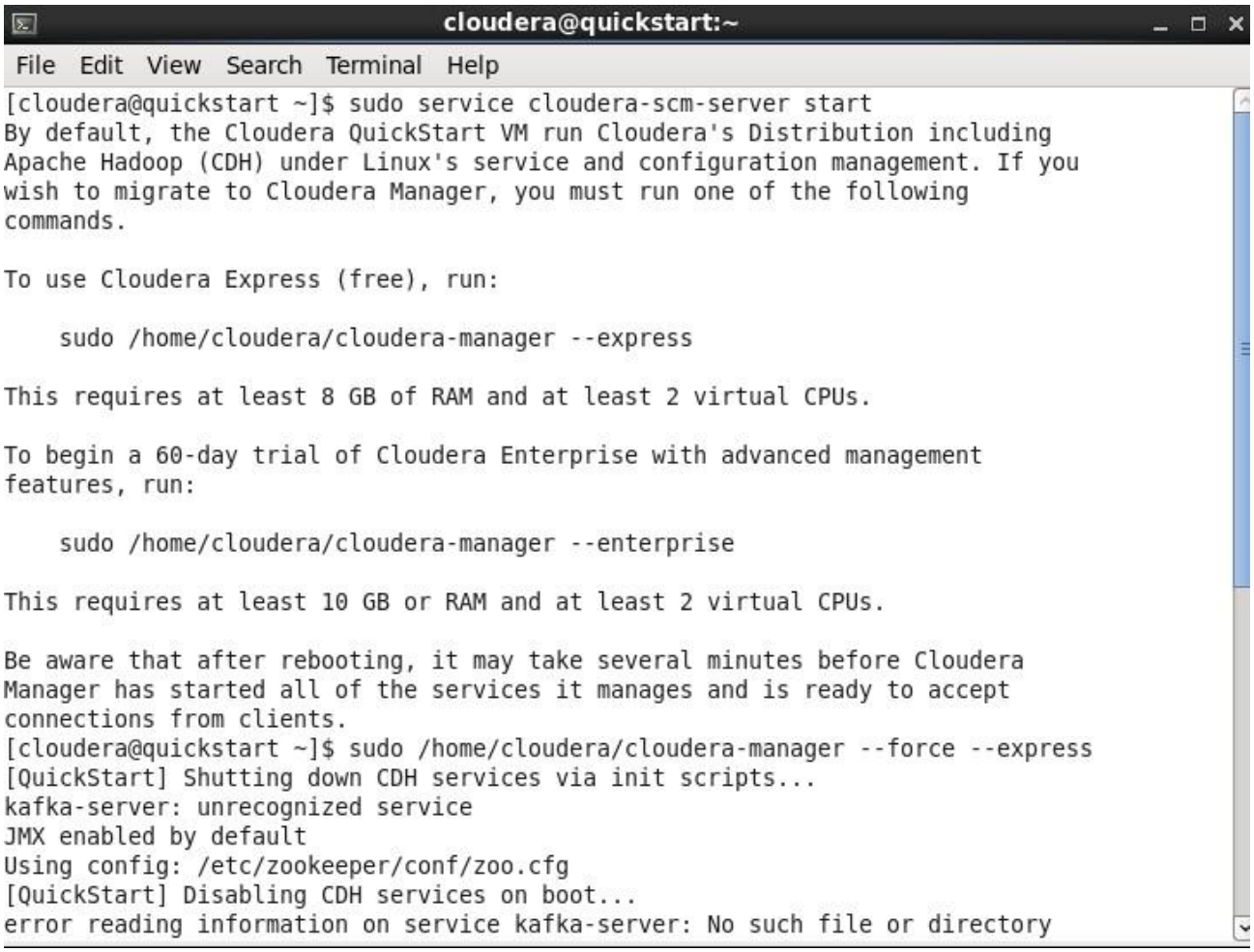
Hadoop. (creating a directory, list the content of directory, upload and download file in HDFS)

- **Prerequisite: dos/linux command**
- **Configure ubantu/cloudera**
- **Test Hadoop services**

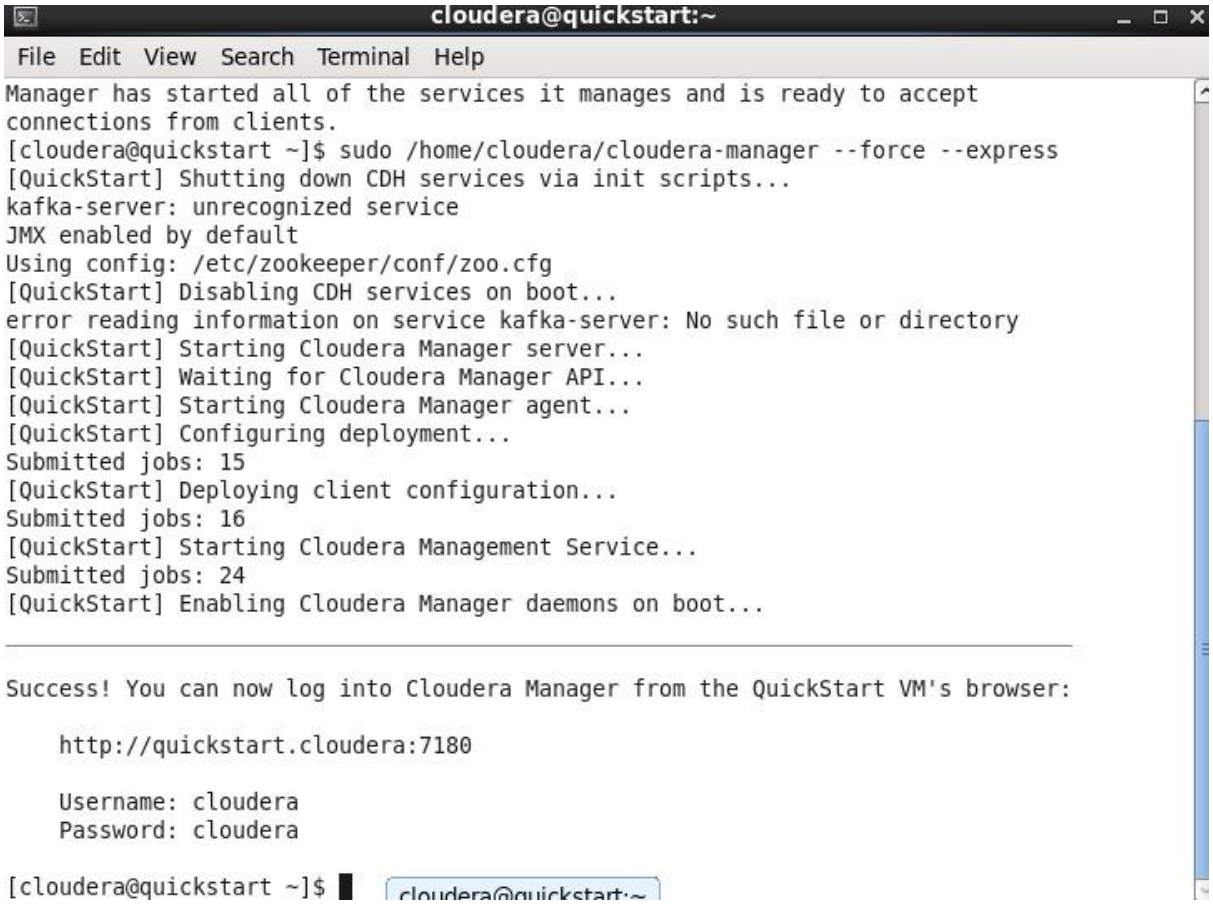
172.16.12.38 - /

7/5/2022	1:04 PM	6578981376	cloudera-quickstart-vm-5.12.0-0-vmware.ova
7/15/2021	4:36 PM	5811085017	cloudera-quickstart-vm-5.12.0-0-vmware.zip
9/26/2019	1:07 PM	536135688	VMware-workstation-full-15.0.4-12990004.exe
9/28/2019	1:26 PM	434232619	VMware_Workstation_14_Pro.zip
7/5/2022	12:59 PM	301	web.config



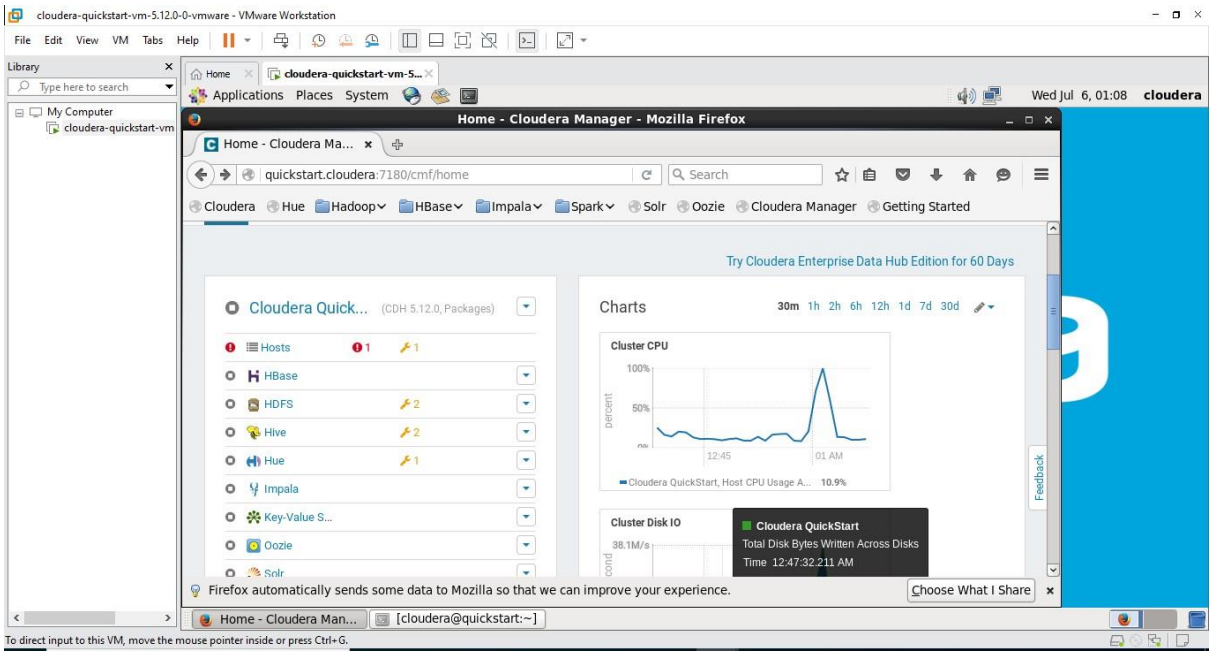


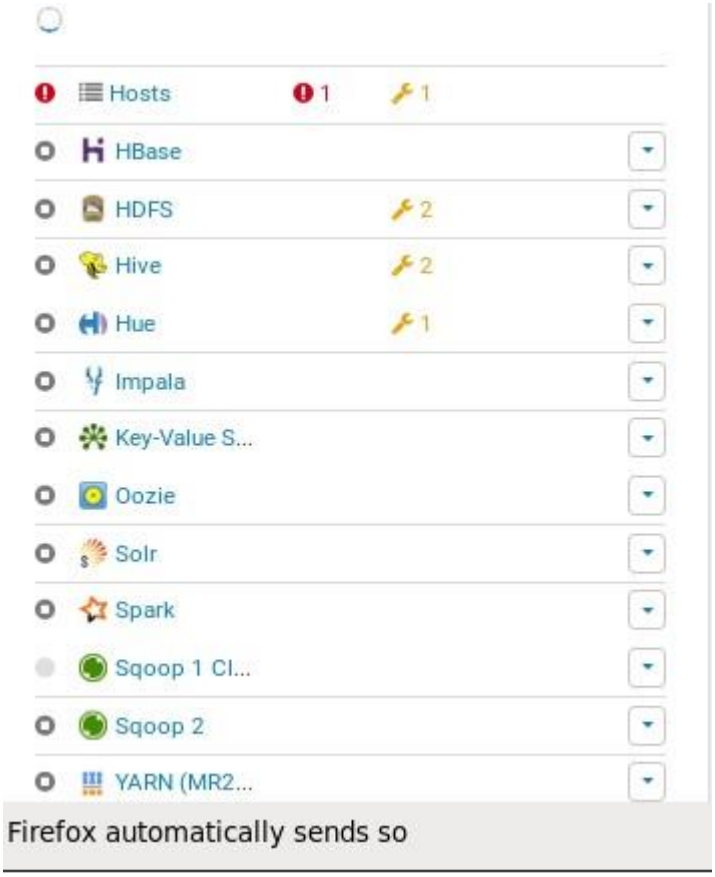
```
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ sudo service cloudera-scm-server start
By default, the Cloudera QuickStart VM run Cloudera's Distribution including
Apache Hadoop (CDH) under Linux's service and configuration management. If you
wish to migrate to Cloudera Manager, you must run one of the following
commands.

To use Cloudera Express (free), run:

    sudo /home/cloudera/cloudera-manager --express

This requires at least 8 GB of RAM and at least 2 virtual CPUs.

To begin a 60-day trial of Cloudera Enterprise with advanced management
features, run:

    sudo /home/cloudera/cloudera-manager --enterprise

This requires at least 10 GB or RAM and at least 2 virtual CPUs.

Be aware that after rebooting, it may take several minutes before Cloudera
Manager has started all of the services it manages and is ready to accept
connections from clients.
[cloudera@quickstart ~]$ sudo /home/cloudera/cloudera-manager --force --express
[QuickStart] Shutting down CDH services via init scripts...
kafka-server: unrecognized service
JMX enabled by default
Using config: /etc/zookeeper/conf/zoo.cfg
[QuickStart] Disabling CDH services on boot...
error reading information on service kafka-server: No such file or directory
```



Login user and password

And check dasboard




Hosts
1
1
HBase
HDFS
2
Hive
2
Hue
1
Impala
Key-Value S...
Oozie
Solr
Spark
Sqoop 1 Cl...
Sqoop 2
YARN (MR2...
Firefox automatically sends so

Practical 2

Aim: Copy your data into the Hadoop Distributed File System (HDFS), creating a directory, list the content of directory, upload and download file in HDFS

- **Prerequisite: dos/linux command**
- **Configure HDFS**

Renaming a file in hdfs:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mv /user/19ce063/arpan.txt /use
r/19ce063/19ce063_arpan.txt
[cloudera@quickstart ~]$ hadoop fs -ls /user/19ce063/
Found 1 items
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:00 /user/19ce063/19ce063
_arpan.txt
```

Copying file in hdfs:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -cp /user/19ce063/19ce063_arpan.txt /user/19c
e063/copy-arpan.txt
22/07/28 00:07:42 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream
.java:952)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:
690)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:879)
[cloudera@quickstart ~]$ hadoop fs -ls /user/19ce063/
Found 2 items
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:00 /user/19ce063/19ce063_arpan.txt
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:07 /user/19ce063/copy-arpan.txt
```

Copying file from another folder:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -cp /user/19ce063/19ce063_arpan.txt /user/cop
ying_in_anotherfolder.txt
[cloudera@quickstart ~]$ hadoop fs -ls /user
Found 12 items
drwxr-xr-x   - 19ce063  supergroup          0 2022-07-28 00:07 /user/19ce063
drwxr-xr-x   - cloudera cloudera            0 2022-07-21 00:50 /user/cloudera
-rw-r--r--   1 hdfs     supergroup         18 2022-07-28 00:10 /user/copying_in_anotherfolder
.txt
drwx------   - hdfs     supergroup          0 2022-07-21 00:57 /user/hdfs
drwxr-xr-x   - hdfs     supergroup          0 2022-07-21 00:32 /user/hduser
drwxr-xr-x   - mapred   hadoop              0 2017-07-19 06:29 /user/history
drwxrwxrwx   - hive     supergroup          0 2017-07-19 06:31 /user/hive
drwxrwxrwx   - hue      supergroup          0 2017-07-19 06:30 /user/hue
drwxrwxrwx   - jenkins  supergroup          0 2017-07-19 06:29 /user/jenkins
drwxrwxrwx   - oozie    supergroup          0 2017-07-19 06:30 /user/oozie
drwxrwxrwx   - root     supergroup          0 2017-07-19 06:29 /user/root
drwxr-xr-x   - hdfs     supergroup          0 2017-07-19 06:31 /user/spark
```

Copying folder into another folder:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /user/dummy
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mv /user/dummy /user/19ce063
[cloudera@quickstart ~]$ hadoop fs -ls /user/19ce063
Found 3 items
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:00 /user/19ce063/19ce063_arpan.txt
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:07 /user/19ce063/copy-arpan.txt
drwxr-xr-x   - hdfs supergroup          0 2022-07-28 00:12 /user/19ce063/dummy
```

Reading copied content file:

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/19ce063/copy-arpan.txt
I am Arpan Ladani
[cloudera@quickstart ~]$
```

To delete trash:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -expunge
22/07/28 00:33:54 INFO fs.TrashPolicyDefault: TrashPolicyDefault#deleteCheckpoint for trashRo
ot: hdfs://quickstart.cloudera:8020/user/hdfs/.Trash
22/07/28 00:33:54 INFO fs.TrashPolicyDefault: TrashPolicyDefault#deleteCheckpoint for trashRo
ot: hdfs://quickstart.cloudera:8020/user/hdfs/.Trash
22/07/28 00:33:54 INFO fs.TrashPolicyDefault: TrashPolicyDefault#createCheckpoint for trashRo
ot: hdfs://quickstart.cloudera:8020/user/hdfs/.Trash
22/07/28 00:33:54 INFO fs.TrashPolicyDefault: Created trash checkpoint: /user/hdfs/.Trash/220
728003354
[cloudera@quickstart ~]$
```

Creating new empty file in hdfs:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -touchz /user/19ce063/new-arpan.txt
[cloudera@quickstart ~]$ hadoop fs -ls /user/19ce063
Found 3 items
-rw-r--r--   1 hdfs supergroup         18 2022-07-28 00:00 /user/19ce063/19ce063_arpan.txt
drwxr-xr-x   - hdfs supergroup          0 2022-07-28 00:12 /user/19ce063/dummy
-rw-r--r--   1 hdfs supergroup          0 2022-07-28 00:35 /user/19ce063/new-arpan.txt
```

To know the size of file:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -du /user/19ce063/
18  18  /user/19ce063/19ce063_arpan.txt
0   0   /user/19ce063/dummy
0   0   /user/19ce063/new-arpan.txt
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -du /user/
18          18          /user/19ce063
18          18          /user/cloudera
18          18          /user/copying_in_anotherfolder.txt
18          18          /user/hdfs
0           0           /user/hduser
0           0           /user/history
0           0           /user/hive
0           0           /user/hue
0           0           /user/jenkins
858119738   858119738   /user/oozie
0           0           /user/root
0           0           /user/spark
```

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -df /user/19ce063
Filesystem                              Size       Used    Available  Use%
hdfs://quickstart.cloudera:8020  58479091712  869449728  46100221952    1%
```

To see statistics of folder:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -stat /user/19ce063
2022-07-28 07:35:45
```

To check health of file or folder:

```
[cloudera@quickstart ~]$ hadoop fsck /user/19ce063
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Connecting to namenode via http://quickstart.cloudera:50070/fsck?ugi=cloudera&path=%2Fuser%2F
19ce063
FSCK started by cloudera (auth:SIMPLE) from /127.0.0.1 for path /user/19ce063 at Thu Jul 28 0
1:07:32 PDT 2022
..Status: HEALTHY
 Total size:    18 B
 Total dirs:    2
 Total files:   2
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 18 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Thu Jul 28 01:07:32 PDT 2022 in 2 milliseconds


The filesystem under path '/user/19ce063' is HEALTHY
```

Change group with a new name (supergroup -> newgroup):

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs chgrp newgroup /user/19ce063/19ce063_arpan.txt
chgrp: Unknown command
Did you mean -chgrp?  This command begins with a dash.
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -chgrp newgroup /user/19ce063/19ce063_arpan.txt
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -ls /user/19ce063
Found 3 items
-rw-r--r--   1 hdfs newgroup          18 2022-07-28 00:00 /user/19ce063/19ce063_arpan.txt
drwxr-xr-x   - hdfs supergroup          0 2022-07-28 00:12 /user/19ce063/dummy
-rw-r--r--   1 hdfs supergroup          0 2022-07-28 00:35 /user/19ce063/new-arpan.txt
```

To merge content of two file into one file:

```
[cloudera@quickstart ~]$ hadoop fs -appendToFile - /user/root/file.txt
Arpan is attending lab.
^C22/07/28 01:26:55 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeRe
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBloc
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFS
[cloudera@quickstart ~]$ hadoop fs -cat /user/root/file.txt
Arpan is attending lab.
```

To check the checksum:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -checksum /user/19ce063/19ce063_arpan.txt
/user/19ce063/19ce063_arpan.txt MD5-of-0MD5-of-512CRC32C        0000020000000000000000006d98a52f51b666f6a85d45f62ec4a668
[cloudera@quickstart ~]$
```

Aim:

Practical 3

To understand the overall programming architecture using Map Reduce API.(word count using Map-Reduce)

Understand flow of MR,

- **Create mapper and reducer file**
- **Set MR execution from CMD over HDFS**

mapper.py

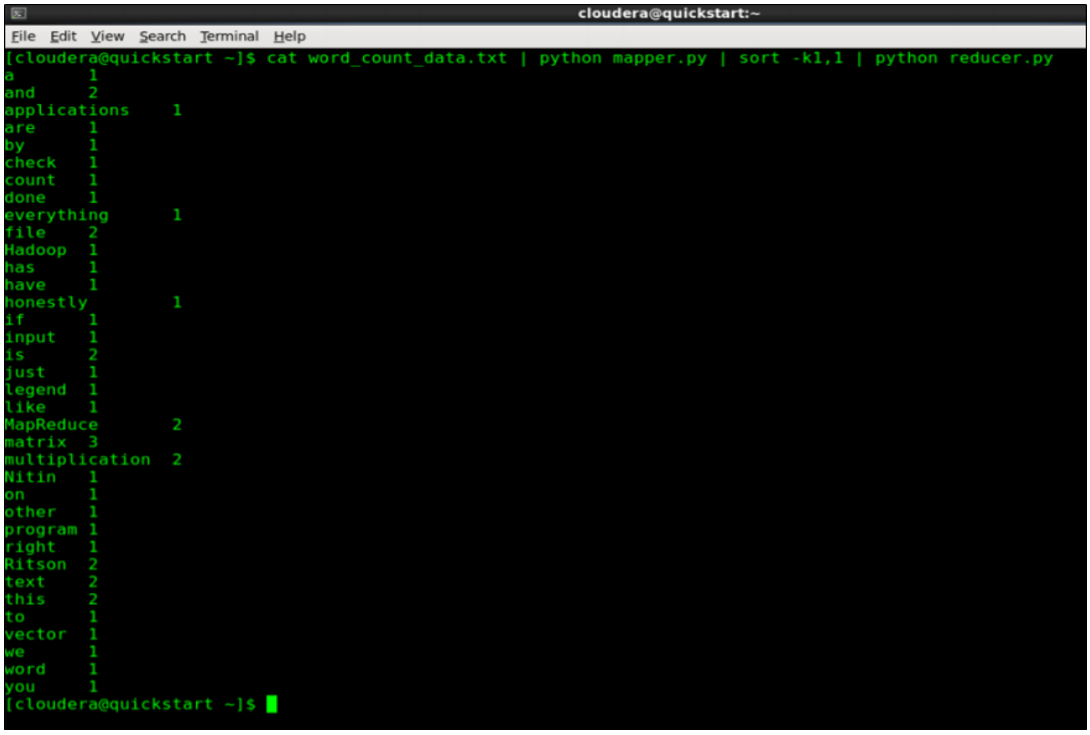
```
#!/usr/bin/python
import sys for line in
sys.stdin:     line =
line.strip()     words
= line.split()     for
word in words:
        print('%s\t%s' % (word, 1))
```

reducer.py

```
#!/usr/bin/python

from operator import itemgetter import
sys

current_word = None
current_count = 0 word = None
for line in sys.stdin:     line =
```

```
line.strip()     word, count =
line.split('\t', 1)     try:
        count = int(count)
except ValueError:
        continue


    if current_word == word:
current_count += count
else:         if current_word:
            print('%s\t%s' % (current_word, current_count))
current_count = count         current_word = word if
current_word == word:
    print('%s\t%s' % (current_word, current_count))
```

➔ **Output**



Aim:

Practical 4

Configure Hive and perform SQL like operation Steps:

1) Open HDFS in Cloudera and create csv file.

```
cloudera@quickstart:~/Documents
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ ls
cloudera-manager  Downloads                  kerberos   parcels    Videos
cm_api.py         eclipse                    lib        Pictures   workspace
Desktop           enterprise-deployment.json Music      Public
Documents         express-deployment.json    myNewFile  Templates
[cloudera@quickstart ~]$ cd Documents
[cloudera@quickstart Documents]$ ls
cloudera-manager.html  Customer.csv   Employee.csv  Order.csv
css                    employee.csv~  img           Order.csv~
[cloudera@quickstart Documents]$ cat Employee.csv
Id,Name,Dept,Yoj,Salary
1,Rose ,IT ,2012,26000
2,Sam,Sales,2012,22000
3,Mike,HR,2013,30000
4,Nick,SC ,2013,20000
[cloudera@quickstart Documents]$
```

2) Open HIVE in command promt

3) Create Database

```
hive> CREATE DATABASE office;
```

4) Use database

```
hive> USE office;
```

5) Create Table

```
office
Time taken: 0.02 seconds, Fetched: 2 row(s)
hive> create table employee
    > (Id INT, Name STRING, Dept STRING, Yoj INT, salary INT)
    > row format delimited fields terminated by ','
    > tblproperties ("skip.header.line.count"="1");
```

6) Insert Data from the created csv file into created table.

```
Time taken: 0.271 seconds
hive> show tables;
OK
employee
Time taken: 0.056 seconds, Fetched: 1 row(s)
hive> describe employee;
OK
id                      int
name                    string
dept                    string
yoj                     int
salary                  int
Time taken: 0.201 seconds, Fetched: 5 row(s)
hive> select * from employee;
OK
Time taken: 0.621 seconds
hive> LOAD DATA LOCAL INPATH
    > '/home/cloudera/Documents/Employee.csv'
    > INTO TABLE employee;
Loading data to table office.employee
Table office.employee stats: [numFiles=1, totalSize=116]
OK
Time taken: 1.147 seconds
hive>
```

7) **Show data and perform different operation.**

Select, Show table, alter table, drop table.

```
hive> select * from office.employee WHERE Salary>25000;
OK
1       Rose    IT      2012    26000
3       Mike    HR      2013    30000
Time taken: 0.296 seconds, Fetched: 2 row(s)
hive> alter table office.employee RENAME TO office.employees;
OK
Time taken: 0.496 seconds
hive> show tables;
OK
employees
Time taken: 0.024 seconds, Fetched: 1 row(s)
hive> drop table employees;
OK
Time taken: 0.363 seconds
hive> show tables;
OK
Time taken: 0.017 seconds
hive>
```

Practical 5

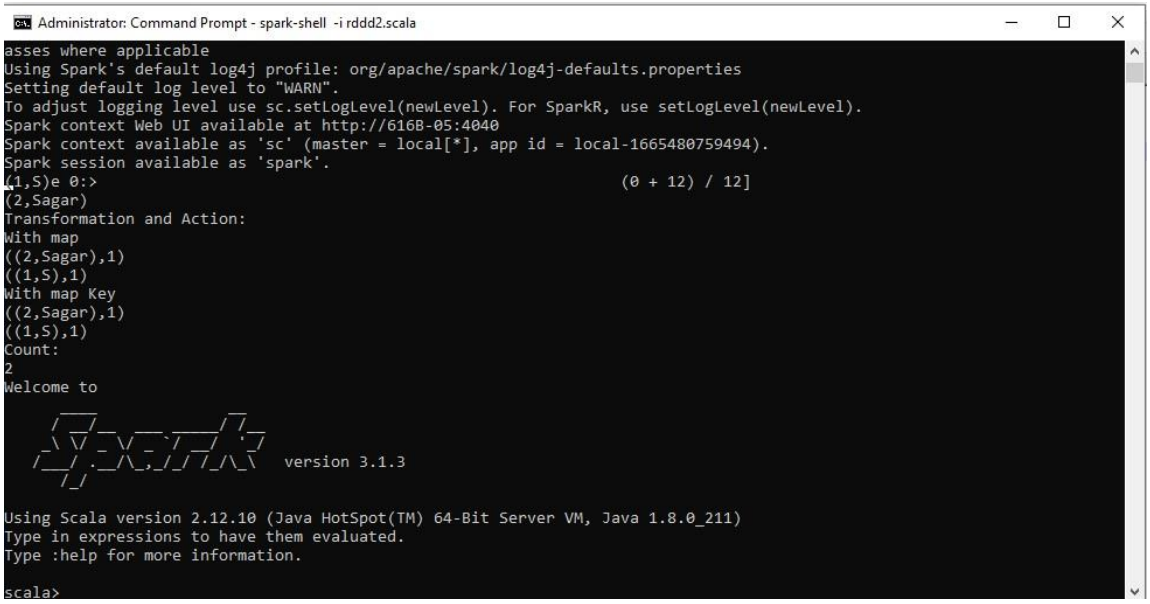
Configure Spark and perform Action and Transformation on RDD

Program:

Aim:

```
val bot = List((1,"S"),(2,"Sagar")) val rdd =
sc.parallelize(bot) rdd.foreach(println)
println("Transformation and Action:")
println("With map") val rdd2=rdd.map(f=>
(f,1)) rdd2.foreach(println) println("With map
Key")
val rdd3=rdd.map(f=> (f,1)).sortByKey()
rdd3.foreach(println) val c = rdd2.count()
println("Count:")
println(c)
```

Output:



Practical 6

Aim: Perform SparkSQL on Netfilx_title dataset(Given) implementation:

```
movies = spark.read.format("csv") \
.option("header", "true") \
  .option("inferSchema", "true") \
  .load("../input/netflix-files-tests/netflix_titles.csv")
```

```
movies.printSchema()
```

```
root
 |-- show_id: string (nullable = true)
 |-- type: string (nullable = true)
 |-- title: string (nullable = true)
 |-- director: string (nullable = true)
 |-- cast: string (nullable = true)
 |-- country: string (nullable = true)
 |-- date_added: string (nullable = true)
 |-- release_year: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- duration: string (nullable = true)
 |-- listed_in: string (nullable = true)
 |-- description: string (nullable = true)
```

```
movies.show(3)
```

```
+-------+-------+-----+-----------------+--------------------+---------+-----------------+------------+------+---------+------
--------------+--------------------+
|show_id|   type|title|         director|                cast|  country|       date_added|release_year|rating| duration|
listed_in|         description|
+-------+-------+-----+-----------------+--------------------+---------+-----------------+------------+------+---------+------
--------------+--------------------+
|     s1|TV Show|   3%|             null|João Miguel, Bian...|   Brazil|  August 14, 2020|        2020| TV-MA|4 Seasons|Intern
ational TV ...|In a future where...|
|     s2|  Movie| 7:19|Jorge Michel Grau|Demián Bichir, Hé...|   Mexico|December 23, 2016|        2016| TV-MA|   93 min|Drama
s, Internati...|After a devastati...|
|     s3|  Movie|23:59|     Gilbert Chan|Tedd Chan, Stella...|Singapore|December 20, 2018|        2011|     R|   78 min|Horror
Movies, In...|When an army recr...|
+-------+-------+-----+-----------------+--------------------+---------+-----------------+------------+------+---------+------
--------------+--------------------+
only showing top 3 rows
```

```
df = movies.select('title', 'release_year', 'country', 'rating') df.show(4)
```

```
+-----+------------+-------------+------+
|title|release_year|      country|rating|
+-----+------------+-------------+------+
|   3%|        2020|       Brazil| TV-MA|
| 7:19|        2016|       Mexico| TV-MA|
|23:59|        2011|    Singapore|     R|
|    9|        2009|United States| PG-13|
+-----+------------+-------------+------+
only showing top 4 rows
```

```
df.printSchema()
```

```
root
 |-- title: string (nullable = true)
 |-- release_year: string (nullable = true)
 |-- country: string (nullable = true)
 |-- rating: string (nullable = true)
```

```
df2 = df.withColumn('year', df['release_year'].cast('int')).drop('release_year
')
df2.printSchema()
```

```
root
 |-- title: string (nullable = true)
 |-- country: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- year: integer (nullable = true)
```

```
df2.filter('year > 2015').show(5)
```

```
+-----+-------+------+----+
|title|country|rating|year|
+-----+-------+------+----+
|   3%| Brazil| TV-MA|2020|
| 7:19| Mexico| TV-MA|2016|
|   46| Turkey| TV-MA|2016|
|  122|  Egypt| TV-MA|2019|
|  706|  India| TV-14|2019|
+-----+-------+------+----+
only showing top 5 rows
```

```
# min e max
from pyspark.sql.functions import max, min
```

```
df2.select(max('year')).show(3)
```

```
+---------+
|max(year)|
+---------+
|     2021|
+---------+
```

```
df2.filter('year == 2021 and country is not NULL').show(10)
```

```
+--------------------+--------------------+------+----+
|               title|             country|rating|year|
+--------------------+--------------------+------+----+
|     Carmen Sandiego|       United States| TV-Y7|2021|
|            Charming|Canada, United St...| TV-Y7|2021|
|           Cobra Kai|       United States| TV-14|2021|
|Crack: Cocaine, C...|       United States| TV-MA|2021|
|      Disenchantment|       United States| TV-14|2021|
| Dream Home Makeover|       United States|  TV-G|2021|
|Headspace Guide t...|       United States|  TV-G|2021|
|               Hilda|United Kingdom, C...| TV-Y7|2021|
|History of Swear ...|       United States| TV-MA|2021|
|Inside the World'...|      United Kingdom| TV-MA|2021|
+--------------------+--------------------+------+----+
only showing top 10 rows
```

Practical 7

Aim: Perform Spark streaming with word count.

Program:

```
val linesDF = sc.textFile("mytext.txt").toDF("line")
val wordsDF = linesDF.explode("line","word")((line: String) => line.split(" "))
val wordCountDF = wordsDF.groupBy("word").count() wordCountDF.show()
```

Input:

```
Sagar IGN is cutebot in bgmi
Sagar IGN is SagarBot in valo
```

Output:

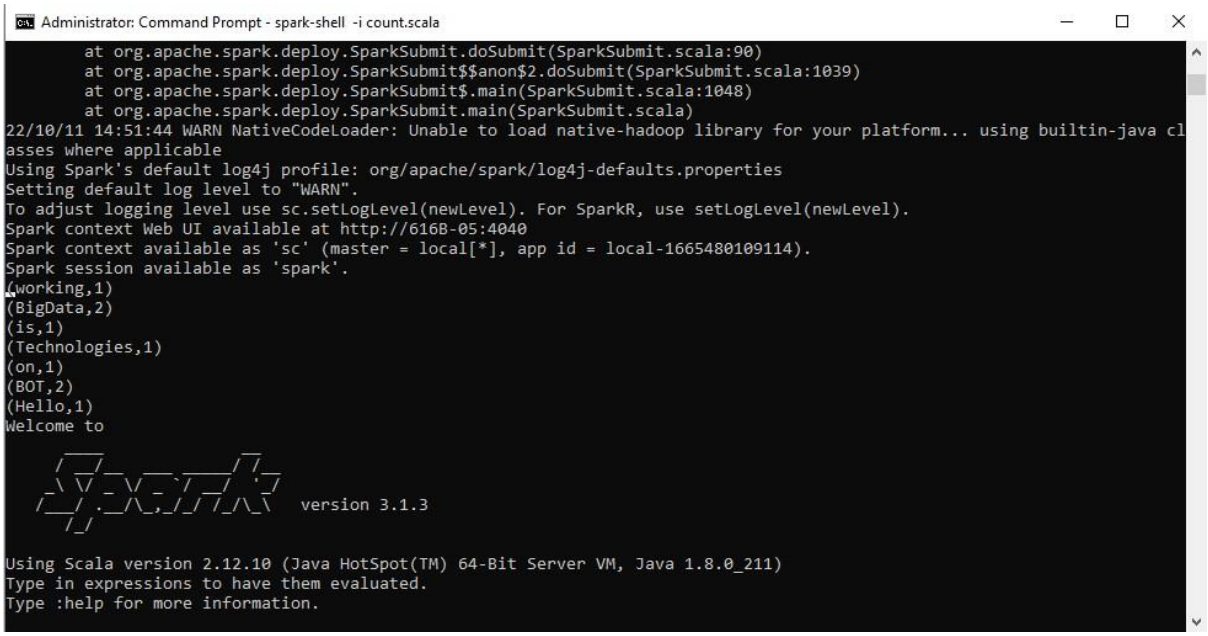
```
Administrator: Command Prompt - spark-shell -i prac7.scala
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://616B-05:4040
Spark context available as 'sc' (master = local[*], app id = local-1665559914480).
Spark session available as 'spark'.
warning: there was one deprecation warning (since 2.0.0); for details, enable `:setting -deprecation' or `:replay -depre
cation'
+--------+-----+
|    word|count|
+--------+-----+
|    bgmi|    1|
|    valo|    1|
|      in|    2|
|      is|    2|
|SagarBot|    1|
|   Sagar|    2|
| cutebot|    1|
|     IGN|    2|
+--------+-----+

Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_211)
Type in expressions to have them evaluated.
Type :help for more information.
```

Practical 8

Aim: Perform word count operation on it using SCALA Program:

```
val list = List("BOT is working on BigData Technologies","Hello BOT","BigData") val
words = list.flatMap(line => line.split(" ")) val keyData = words.map(word =>
(word,1)) val groupedData = keyData.groupBy(_._1) val result =
groupedData.mapValues(list=>{ list.map(_._2).sum
})
result.foreach(println)
```

Output:



Practical – 9

➔ **AIM** : Configure Neo4j and create node(add –remove property) & relationship in it.

➔ **Steps to Configure Neo4J** :

1.Download Neo4J Community Edition from below link: https://neo4j.com/download-center/

2.Extract the zip file and keep the extracted folder in D:\drive (for e.g.)

3.Open command prompt and navigate to D:\neo4j

4.Run below commands to start neo4j service:bin\neo4j installservicebin\neo4j start

5.Open the URL http://localhost:7474/browser/ in your web browser.

6.Initial username: neo4j and password: neo4j

7.Once submitted prompt for change password will be shown wherein you can change database password

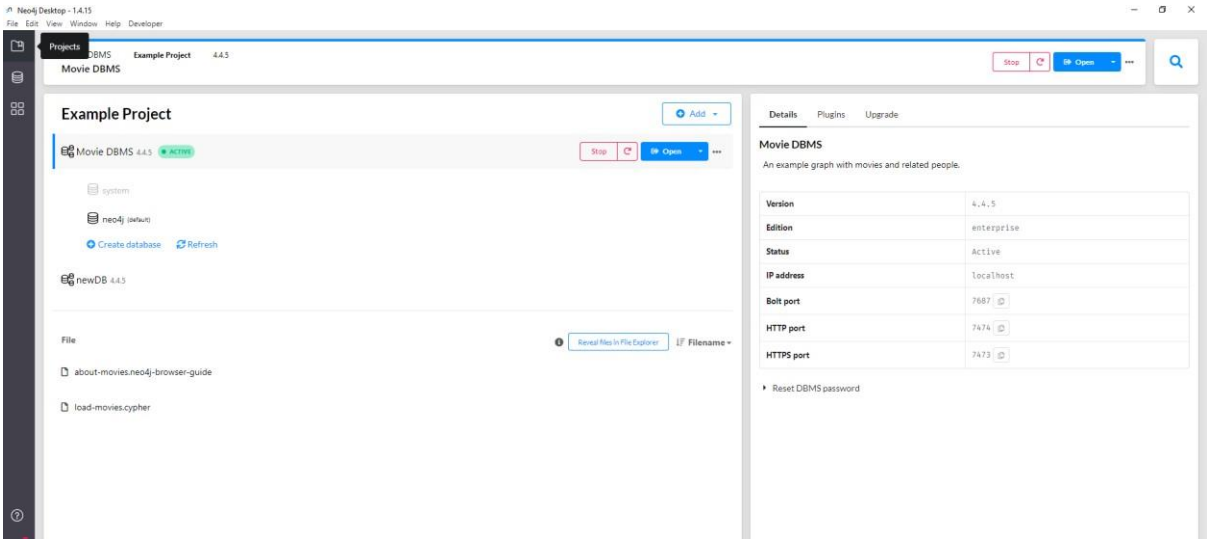
8.Homepage will be shown in which top most text box starting with $ is for writing queries.

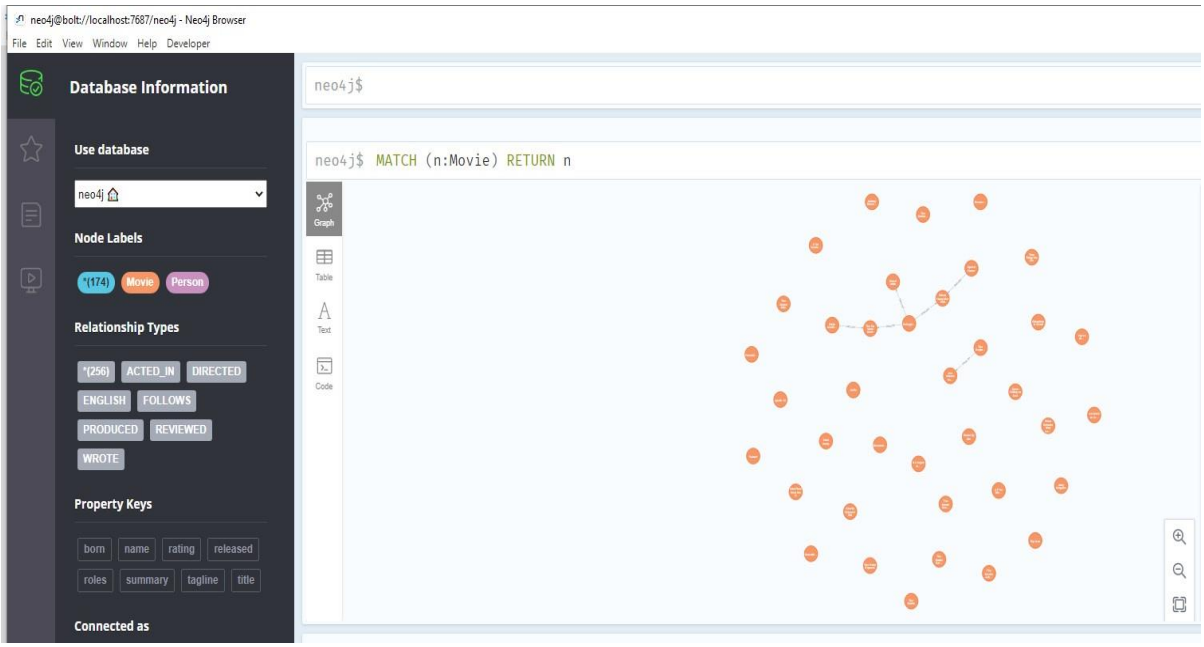
➔ **Code sample** :

(for implementation)

```
CREATE (n: Person {name:"Brad"}) RETURN n
CREATE (n: Person {name:"Alice"}) RETURN n
CREATE (n: Person {name:"Mike"}) RETURN n
CREATE (n: Person {name:"Jill"}) RETURN n
CREATE (n: Person {name:"Hazel"}) RETURN n
CREATE (n: Movie  {title:"Unforgiven "}) RETURN n
CREATE (n: Movie {title:"The Da Vinci Code"}) RETURN n
CREATE (n: Movie {title:"When Harry Met Sally"}) RETURN n
```

➔ **Output Screenshots :**


Neo4j Desktop - 1.4.15
File Edit View Window Help Developer
Projects
Example Project
Movie DBMS
Stop
Open
Add
Details
Plugins
Upgrade
Movie DBMS 4.4.5 ACTIVE
system
neo4j (default)
Create database
Refresh
newDB 4.4.5
File
Reveal files in File Explorer
Filename
about-movies.neo4j-browser-guide
load-movies.cypher
An example graph with movies and related people.
Version 4.4.5
Edition enterprise
Status Active
IP address localhost
Bolt port 7687
HTTP port 7474
HTTPS port 7473
Reset DBMS password


neo4j@bolt://localhost:7687/neo4j - Neo4j Browser
File Edit View Window Help Developer
Database Information
Use database
neo4j
Node Labels
*(174) Movie Person
Relationship Types
*(256) ACTED_IN DIRECTED
ENGLISH FOLLOWS
PRODUCED REVIEWED
WROTE
Property Keys
born name rating released
roles summary tagline title
Connected as
neo4j$
neo4j$ MATCH (n:Movie) RETURN n
Graph
Table
Text
Code


neo4j$ MATCH (n:Movie) RETURN n
Graph
Table
Text
Code
Server version Neo4j/4.4.5
Server address localhost:7687
Query MATCH (n:Movie) RETURN n
Summary {, "query": {, "text": "MATCH (n:Movie) RETURN n", ...
Response [, {, "keys": [...

Practical – 10

➔ **AIM** : Import files from Neo4j and complete relational graph from it.

➔ **Code sample:**
(for implementation)

1.To add a relationship to nodes:

```
MATCH (a:Person{name:"Brad"}),(b:Person{name:"Alice"}) MERGE (a) -[r:FRIENDS]-> (b)
```

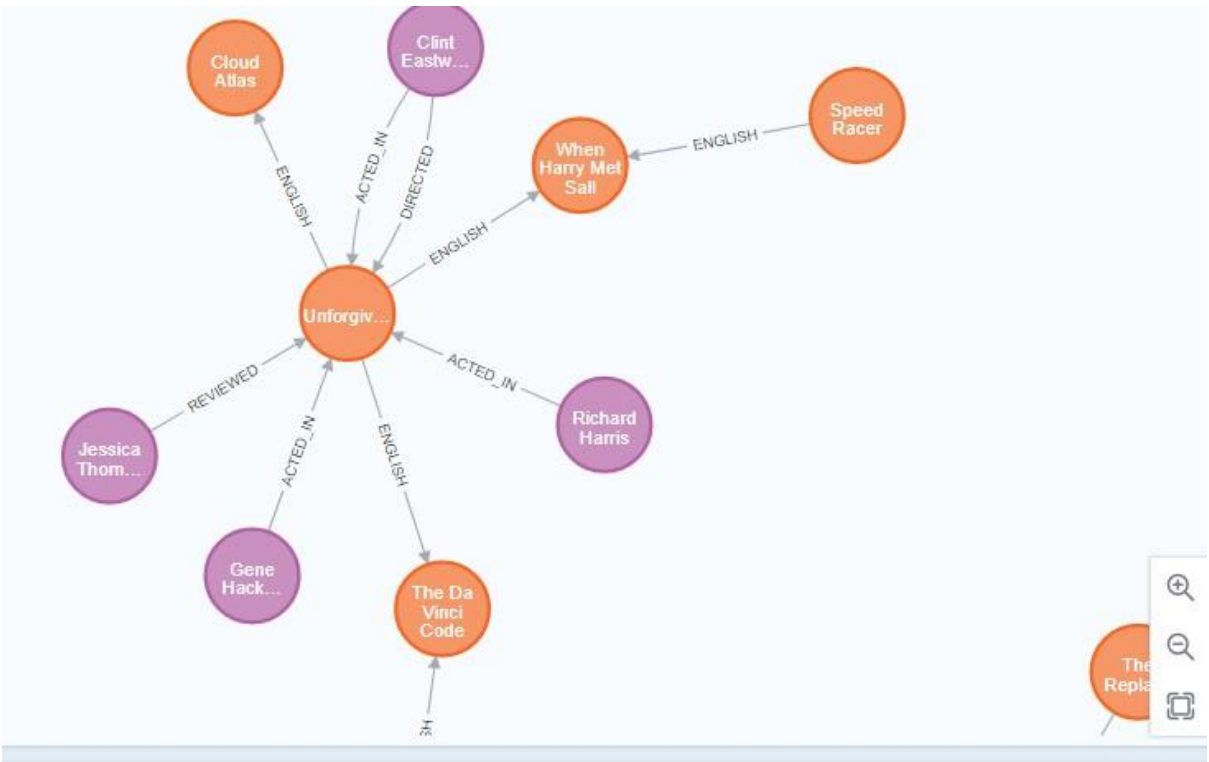
2.To add properties in relationship:

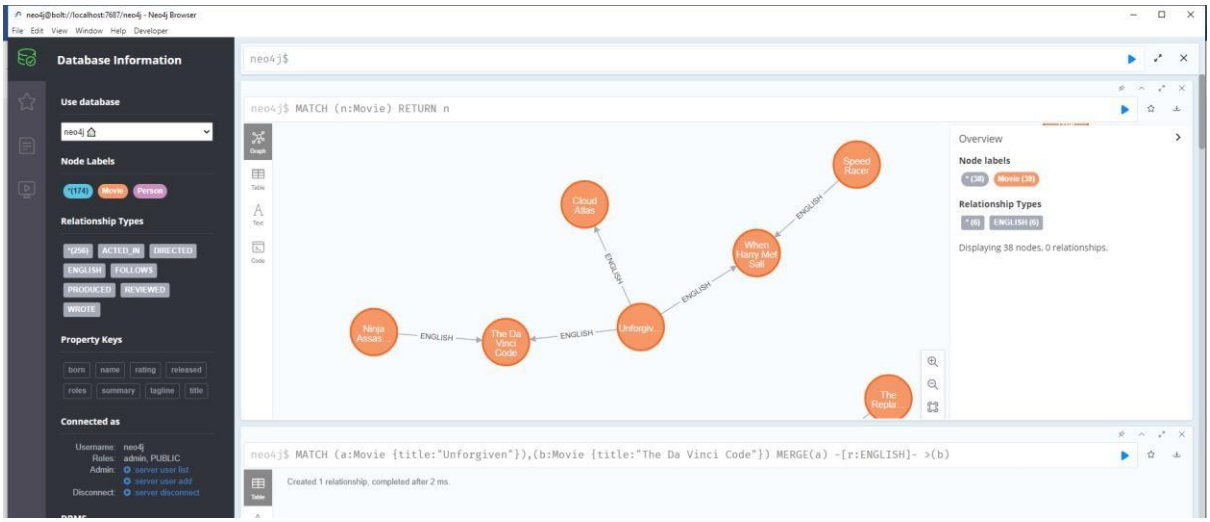
```
MATCH (a: Person {name:"Brad"}),(b: Person {name:"MIKE"}) MERGE (a) -[r: FRIENDS
{since:"1998"}]-> (b)
```

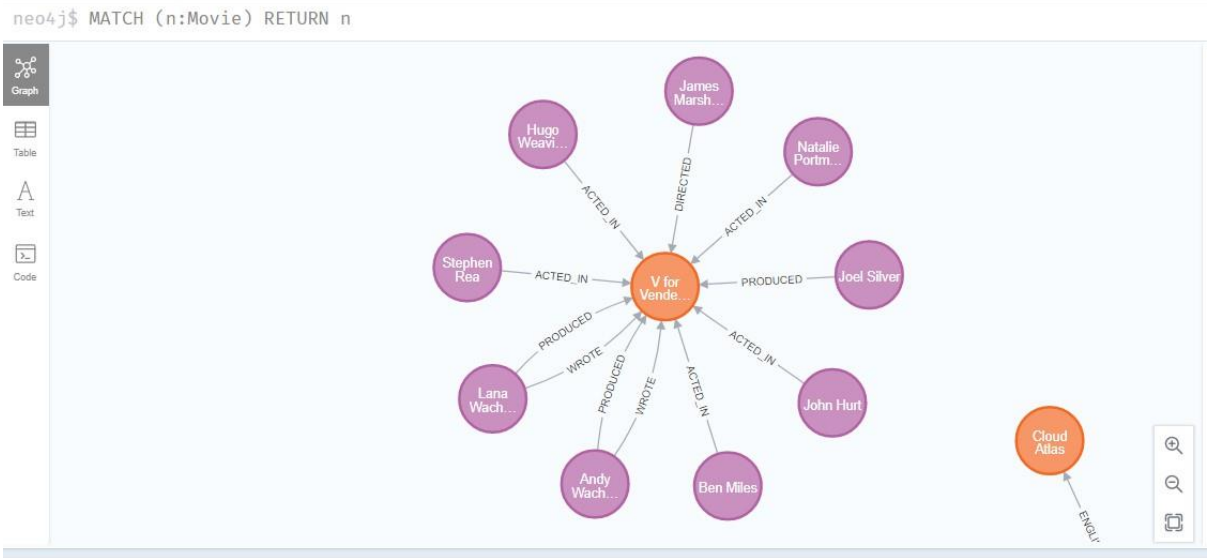
3.Adding relationship between two different labels:

```
MATCH (a:Person{name:"Jill"}),(b:Movie{name:"Avengers"}) MERGE (a) -[r:FAVOURITE]-> (b)
```

➔ **Implementation Screenshots** :




neo4j@bolt://localhost:7687/neo4j - Neo4j Browser
Database Information
Use database
Node Labels
Relationship Types
Property Keys
Connected as
neo4j$ MATCH (n:Movie) RETURN n
Overview
Node labels
Relationship Types
Displaying 38 nodes, 0 relationships.
Cloud Atlas
Speed Racer
When Harry Met Sall
Ninja Assas...
The Da Vinci Code
Unforgiv...
ENGLISH
neo4j$ MATCH (a:Movie {title:"Unforgiven"}),(b:Movie {title:"The Da Vinci Code"}) MERGE(a) -[r:ENGLISH]- >(b)
Created 1 relationship, completed after 2 ms.


neo4j$ MATCH (n:Movie) RETURN n
Graph
Table
Text
Code
James Marsh...
Hugo Weavi...
Natalie Portm...
Stephen Rea
V for Vende...
Joel Silver
Lana Wach...
John Hurt
Andy Wach...
Ben Miles
Cloud Atlas
ACTED_IN
DIRECTED
PRODUCED
WROTE

A	B
C	D

E	F

Description

Table 1: Name of table

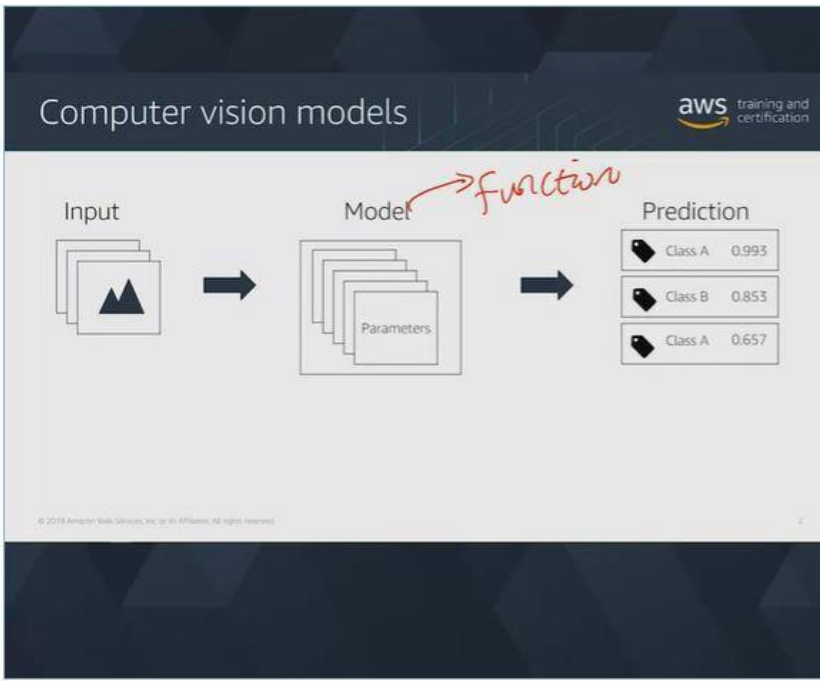


Figure 1: Name of image/figure/diagram