University of Mumbai

PRACTICAL JOURNAL

514

Big Data Systems

SUBMITTED BY

Harsh Lata

SEAT NO :_____

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR QUALIFYING **M.S (Cybersecurity) PART-I (SEMESTER – II)**

EXAMINATION

2024-2025

DEPARTMENT OF INFORMATION TECHNOLOGY 3RD FLOOR,
DR. SHANKAR DAYAL SHARMA BHAVAN, IDOL BUILDING,
VIDYANAGRI, SANTACRUZ (E),
MUMBAI – 400098.

University of Mumbai

Department of Information Technology

Certificate

This is to certify that MS.. **Harsh Lata** ,

Seat No._____Studying in M.S (Cybersecurity) **Part I Semester II** has satisfactorily completed the Practical of **514 Big Data Systems** as prescribed by University of Mumbai, during the academic year **2024 - 25**.

_____ _____

Signature
Subject-In-Charge

Signature
Head of the Department

_____

Signature External
Examiner

College Seal:_____ **Date:** _____

Index

No	Title of Practical	Date	Pg No.	Signature
1	Install, configure and run Hadoop and HDFS	25/02/25	1-8	
2	File Management tasks in Hadoop File System	04/02/25	9-10	
3	Implement word count program using MapReduce	06/03/25	11-12	
4	Install, configure and run Pig. Execute Pig Latin scripts to sort, group, join, project and filter data.	07/03/25	13-17	
5	Install, configure and run Hive.	11/03/25	18-21	
6	Implement Bucketing using Hive	20/03/25	22-25	
7	Install, configure and run Apache Spark.	25/03/25	26-29	
8	Install MongoDB and manipulate it using Python	27/03/25	30-33	
9	Install, configure and run Apache Strom	01/04/25	34-37	
10	Install, configure and run Apache Solr	04/04/25	38-41	

Practical 1

Aim: Install, configure & run Hadoop and HDFS on Ubuntu (Basic).

Pre-Requisites: An Ubuntu server VM with a user having sudo privileges

Code:

 # Check existing users on ubuntu

cut -d: -f1 /etc/passwd

```
udit@ubuntu:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
```

```
pulse
gnome-initial-setup
gdm
udit
systemd-coredump
sshd
udit@ubuntu:~$
```

 # Remove hadoop user

sudo deluser hduser

```
udit@ubuntu:~$ sudo deluser hduser
[sudo] password for udit:
Removing user `hduser' ...
Warning: group `hadoop' has no more members.
Done.
```

 # ps aux | grep

sudo killall -TERM -u hduser

```
udit@ubuntu:~$ sudo killall -TERM -u hduser
Cannot find user hduser
```

 # Remove hadoop group

sudo deluser --group hadoop

```
udit@ubuntu:~$ sudo deluser --group hadoop
Removing group `hadoop' ...
Done.
```

 # Check presence of hadoop

 # Go to location /usr/local/

 # If you see a hadoop folder then hadoop installation was attempted and needs to be removed before fresh installation

 # Remove hadoop

sudo rm -r -f /usr/local/hadoop/

```
udit@ubuntu:~$ sudo rm -r -f /usr/local/hadoop/
```

Step 1 — Installing Java

Step 2 — Installing Hadoop

Step 3 — Configuring Hadoop

Step 4 — Running Hadoop

Step 1 — Installing Java

#To get started, we'll update our package list:

sudo apt update

```
udit@ubuntu:~$ sudo apt update
[sudo] password for udit:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRel
ease
Get:2 http://security.ubuntu.com/ubuntu focal-securit
y InRelease [114 kB]
```

#Next, we'll install OpenJDK, the default Java Development Kit on Ubuntu 18.04:

sudo apt install default-jdk

```
udit@ubuntu:~$ sudo apt install default-jdk
Waiting for cache lock: Could not get lock /var/lib/d
pkg/lock-frontend. It is held by process 3431 (unatte
Waiting for cache lock: Could not get lock /var/lib/d
pkg/lock-frontend. It is held by process 3431 (unatte
```

 # Check folder for java installation at location -

 # Java path -/usr/lib/jvm/java-11-openjdk-amd64

 # Once the installation is complete, let's check the version.

java -version

```
udit@ubuntu:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0
ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubu
ntu2.20.04, mixed mode, sharing)
```

This output verifies that OpenJDK has been successfully installed.

#Add new user to new user group

 # group-hadoop, # user - hduser

sudo addgroup hadoop

```
udit@ubuntu:~$ sudo addgroup hadoop
[sudo] password for udit:
Adding group `hadoop' (GID 1001) ...
Done.
```

sudo adduser --ingroup hadoop hduser

```
udit@ubuntu:~$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' .
..
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

 # Add new user to listed groups

sudo usermod -aG sudo hduser

```
udit@ubuntu:~$ sudo usermod -aG sudo hduser
```

 # Change to new user

su hduser

```
udit@ubuntu:~$ sudo usermod -aG sudo hduser
udit@ubuntu:~$ su hduser
Password:
hduser@ubuntu:/home/udit$
```

 # The prompt shoud look like this - hduser@ubuntu:/

==========#

Create a ssh keygen for the user.

ssh-keygen -t rsa -P " -f ~/.ssh/id_rsa

```
hduser@ubuntu:/home/udit$ ssh-keygen -t rsa -P '' -f
~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.s
sh/id_rsa
Your public key has been saved in /home/hduser/.ssh/i
d_rsa.pub
The key fingerprint is:
SHA256:q0umkF+eoJvwGBbuIDYj0fjpYtpMvgMk0h6IRGdZR6o hd
user@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
|.. oo..o         |
| .o.  o          |
|oo   .           |
|=+o .            |
|*o.E    S        |
|oooo     .       |
|*B* . + .        |
|O%+= B o         |
|=o@+o =.         |
+----[SHA256]-----+
```

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

chmod 0600 ~/.ssh/authorized_keys

```
hduser@ubuntu:/home/udit$ cat ~/.ssh/id_rsa.pub >> ~/
.ssh/authorized_keys
hduser@ubuntu:/home/udit$ chmod 0600 ~/.ssh/authorize
d_keys
```

 # Disable IPv6.

sudo nano /etc/sysctl.conf

```
hduser@ubuntu:/home/udit$ sudo nano /etc/sysctl.conf
[sudo] password for hduser:
```

#add the following lines to the end of the file

net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1

```
###################################################>
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq funct>
# See https://www.kernel.org/doc/html/latest/admin-g>
# for what other values do
#kernel.sysrq=438

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

==========

Step 2 — Downloading & Installing Hadoop

==========

#Extract Hadoop and move to group hadoop

cd /usr/local

sudo tar xvf /home/furkan/Downloads/hadoop-3.2.3.tar.gz
sudo mv hadoop-3.2.3 hadoop
sudo chown -R hduser: hadoop hadoop

```
hduser@ubuntu:/home/udit/Downloads$ sudo mv hadoop-3.
2.3 hadoop
hduser@ubuntu:/home/udit/Downloads$ sudo chown -R hdu
ser: hadoop hadoop
```

==========

#Now open $HOME/.bashrc

sudo nano $HOME/.bashrc

```
hduser@ubuntu:/home/udit/Downloads$ sudo nano $HOME/.
bashrc
```

 # Add the following lines

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_HDFS_HOME=$HADOOP_HOME

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

 # Save file - Ctrl + s
 # Close file - Ctrl + x

 # Run the following command to make changes through the .bashrc file.

source ~/.bashrc

```
hduser@ubuntu:/home/udit/Downloads$ source ~/.bashrc
```

 # Check version of java and hadoop

Command: java -version

```
hduser@ubuntu:/home/udit/Downloads$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0
ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubu
ntu2.20.04, mixed mode, sharing)
```

Command: hadoop version

```
Hadoop 3.2.3
Source code repository https://github.com/apache/hado
op -r abe5358143720085498613d399be3bbf01e0f131
Compiled by ubuntu on 2022-03-20T01:18Z
Compiled with protoc 2.5.0
From source with checksum 39bb14faec14b3aa25388a6d7c3
45fe8
This command was run using /usr/local/hadoop/share/ha
doop/common/hadoop-common-3.2.3.jar
```

==========

#Create a tmp folder in /app/hadoop/tmp and change the owner to hduser.

cd /usr/local

sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop /app/hadoop/tmp/

```
hduser@ubuntu:/usr/local$ cd /usr/local
hduser@ubuntu:/usr/local$ sudo mkdir -p /app/hadoop/tmp
hduser@ubuntu:/usr/local$ sudo chown hduser:hadoop /app/hadoop/tmp/
```

==========

Step 3 — Configuring Hadoop

==========

 # Hadoop requires that you set the path to Java, either as an environment variable or in the Hadoop configuration file.hadoop-env.sh

cd /usr/local/hadoop/etc/hadoop/

```
hduser@ubuntu:/usr/local$ cd /usr/local/hadoop/etc/hadoop/
```

#To Configure Hadoop's Java Home, begin by opening hadoop-env.sh

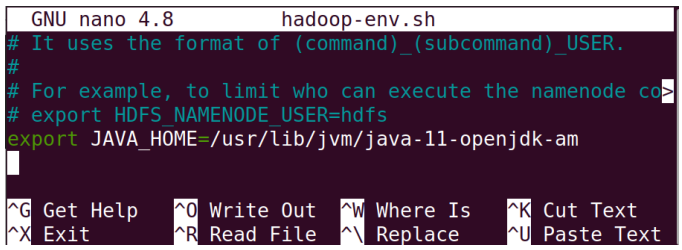
sudo nano hadoop-env.sh

```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano hadoop-env.sh
```

 # Add the following line at the end of .sh file

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

Save & Close



 # Make the changes in core-site.xml file

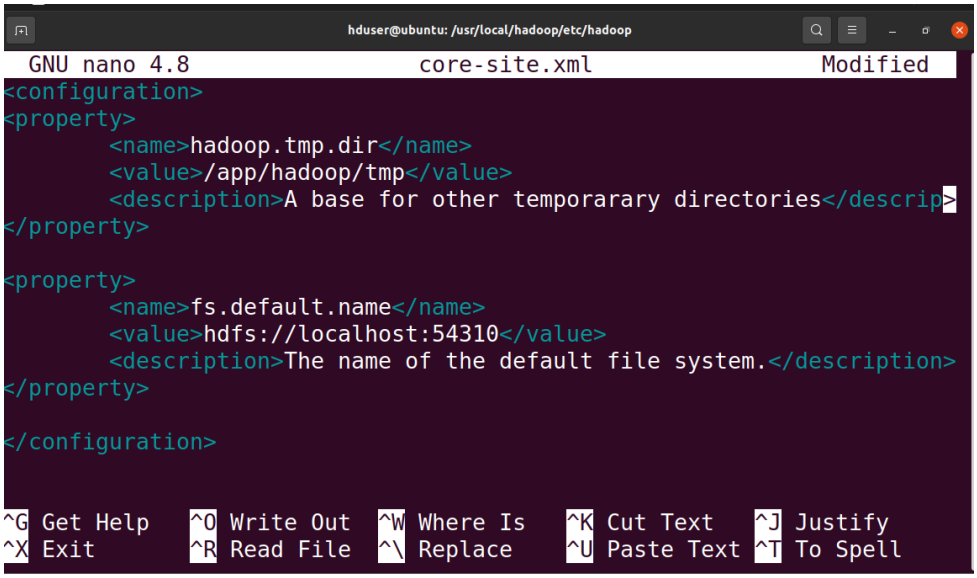
cd /usr/local/hadoop/etc/hadoop

sudo nano core-site.xml

```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cd /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano core-site.xml
```

#Add the following lines

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
        <description>A base for other temporarary directories</description>
</property>
<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:54310</value>
        <description>The name of the default file system.</description>
</property>
```



save & Close

==========

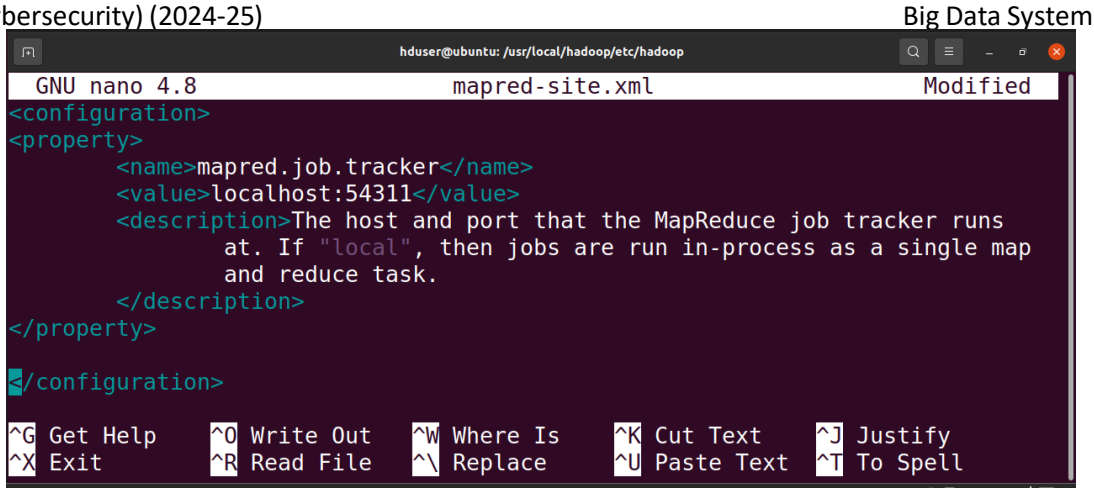
#Make the changes in mapred-site.xml

sudo nano mapred-site.xml

```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano mapred-site.xml
```

#Add the following lines

```
<property>
        <name>mapred.job.tracker</name>
        <value>localhost:54311</value>
        <description>The host and port that the MapReduce job tracker
                runs at. If "local", then jobs are run in-process as a single
                map and reduce task.
        </description>
</property>
```



save & Close

#Make the changes in hdfs-site.xml

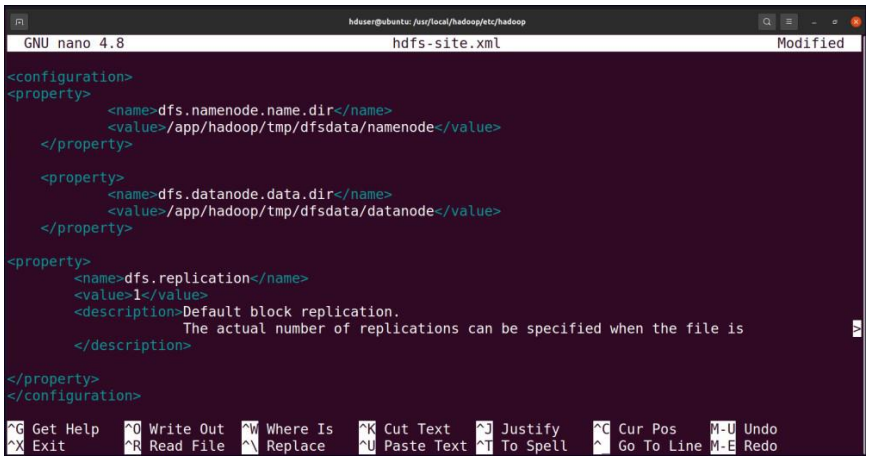
---

sudo nano hdfs-site.xml

```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano
 hdfs-site.xml
```

#Add the following lines

```
<property>
        <name>dfs.namenode.name.dir</name>
        <value>/app/hadoop/tmp/dfsdata/namenode</value>
    </property>

    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/app/hadoop/tmp/dfsdata/datanode</value>
    </property>
<property>
        <name>dfs.replication</name>
        <value>1</value>
        <description>Default block replication.
                The actual number of replications can be specified when the file is
            created. The default is used if replication is not specified in
create time.
        </description>
</property>
```



save & Close

 # Format namenode

hdfs namenode -format

hdfs datanode -format

Step 4 - Running Hadoop

 # To start hadoop, we need to start localhost

ssh localhost

 # If connection is refused, it will result in error - run only if error & then run previous command

sudo apt-get install ssh

 # Start all the hadoop services

/usr/local/hadoop/sbin/start-all.sh

```
hduser@ubuntu:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds
.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
ubuntu: Warning: Permanently added 'ubuntu' (ECDSA) to the list of known hosts
.
Starting resourcemanager
Starting nodemanagers
```

 # Check if that all hadoop services are running (6 services should appear)

jps

```
hduser@ubuntu:~$ jps
69266 DataNode
69126 NameNode
69526 SecondaryNameNode
70059 ResourceManager
70235 NodeManager
70511 Jps
```

 # Access localhost:9870 to get namenode status, open browser and type

http://localhost:9870

 # Stop all the hadoop services.

/usr/local/hadoop/sbin/stop-all.sh

```
hduser@ubuntu:~$ /usr/local/hadoop/sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [ubuntu]
Stopping nodemanagers
Stopping resourcemanager
hduser@ubuntu:~$
```

Conclusion: The performed program to Install, configure & run Hadoop and HDFS on Ubuntu (Basic) of Hadoop on Ubuntu has been successfully demonstrated.

Practical 2

Aim: File Management tasks in Hadoop File System

Start the hadoop and verify all services are started

/usr/local/hadoop/sbin/start-all.sh

```
hduser@paradox:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [paradox]
Starting resourcemanager
Starting nodemanagers
hduser@paradox:~$
hduser@paradox:~$ jps
3889 DataNode
4082 SecondaryNameNode
4852 Jps
3684 NameNode
4486 NodeManager
4360 ResourceManager
```

Create a folder and text file in it on LocalStorage

```
hduser@paradox:~$ mkdir Source_Dir
hduser@paradox:~$ cd Source_Dir/
hduser@paradox:~/Source_Dir$ nano nyfile.txt
hduser@paradox:~/Source_Dir$ cat nyfile.txt
Hadoop is like a messy roommate
– throws data everywhere but somehow,
knows exactly where everything is! 😂
hduser@paradox:~/Source_Dir$
```

Create a folder in HDFS root directory and verify

hdfs dfs –mkdir /destination

hdfs dfs –ls ./

```
hduser@paradox:~/Source_Dir$
hduser@paradox:~/Source_Dir$ hdfs dfs -mkdir /destination
hduser@paradox:~/Source_Dir$ hdfs dfs -ls /
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2025-03-12 23:37 /Media
drwxr-xr-x   - hduser supergroup          0 2025-03-17 23:31 /destination
hduser@paradox:~/Source_Dir$
```

Now move the file from local storage to HDFS

hdfs dfs -copyFromLocal ./nyfile.txt /destination

hdfs dfs -ls /destination

```
hduser@paradox:~/Source_Dir$
hduser@paradox:~/Source_Dir$ hdfs dfs -copyFromLocal ./nyfile.txt /destination
hduser@paradox:~/Source_Dir$ hdfs dfs -ls /destination
Found 1 items
-rw-r--r--   1 hduser supergroup        113 2025-03-17 23:34 /destination/nyfile.txt
hduser@paradox:~/Source_Dir$
```

Read the file from HDFS

hdfs dfs -cat /destination/nyfile.txt

```
hduser@paradox:~/Source_Dir$
hduser@paradox:~/Source_Dir$ hdfs dfs -cat /destination/nyfile.txt
Hadoop is like a messy roommate
– throws data everywhere but somehow,
knows exactly where everything is! 😂
hduser@paradox:~/Source_Dir$
```

Conclusion: The practical to study File Management tasks in Hadoop File System was successfully executed.

Practical 3

Aim: Implement word count / frequency programs using MapReduce

Start the hadoop and verify all services are started

/usr/local/hadoop/sbin/start-all.sh

```
hduser@paradox:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [paradox]
Starting resourcemanager
Starting nodemanagers
hduser@paradox:~$
hduser@paradox:~$ jps
3889 DataNode
4082 SecondaryNameNode
4852 Jps
3684 NameNode
4486 NodeManager
4360 ResourceManager
```

Create a text file

```
hduser@paradox:~/BDS_Prac_3$ nano myfile.txt
hduser@paradox:~/BDS_Prac_3$ cat myfile.txt

meow meow 🐱
wi wi wi
wi wi wi
uyaya uyaya
hduser@paradox:~/BDS_Prac_3$
```

Now move the file from local storage to HDFS

hdfs dfs -put ./myfile.txt /

```
hduser@paradox:~/BDS_Prac_3$ hdfs dfs -put ./myfile.txt /
```

Now, run the mapReduce for WordCount for the file

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar wordcount /myfile.txt /output

```
hduser@paradox:~/BDS_Prac_3$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-
3.2.3.jar wordcount /myfile.txt /output
2025-03-26 23:28:09,377 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-03-26 23:28:09,718 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-03-26 23:28:09,722 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-03-26 23:28:10,359 INFO input.FileInputFormat: Total input files to process : 1
2025-03-26 23:28:10,576 INFO mapreduce.JobSubmitter: number of splits:1
2025-03-26 23:28:11,394 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1019871845_0001
2025-03-26 23:28:11,402 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-03-26 23:28:11,743 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-03-26 23:28:11,749 INFO mapreduce.Job: Running job: job_local1019871845_0001
2025-03-26 23:28:11,844 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-03-26 23:28:12,013 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-03-26 23:28:12,016 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folder
s under output directory:false, ignore cleanup failures: false
2025-03-26 23:28:12,026 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.outp
ut.FileOutputCommitter
2025-03-26 23:28:12,258 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-03-26 23:28:12,263 INFO mapred.LocalJobRunner: Starting task: attempt_local1019871845_0001_m_000000_0
```

Check output at default output location

hdfs dfs -head /output/part-r-00000

```
hduser@paradox:~/BDS_Prac_3$ hdfs dfs -head /output/part-r-00000
meow    2
uyaya   2
wi      6
🐱      1
hduser@paradox:~/BDS_Prac_3$
hduser@paradox:~/BDS_Prac_3$
```

To get output in a .txt file in HDFS

hdfs dfs -mv /output/part-r-00000 /output/opt.txt

```
hduser@paradox:~/BDS_Prac_3$ hdfs dfs -mv /output/part-r-00000 /output/opt.txt
hduser@paradox:~/BDS_Prac_3$ hdfs dfs -ls /output
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2025-03-26 23:39 /output/_SUCCESS
-rw-r--r--   1 hduser supergroup         27 2025-03-26 23:39 /output/opt.txt
hduser@paradox:~/BDS_Prac_3$
```

Check the file system in HDFS

hdfs dfs -ls /

```
hduser@paradox:~/BDS_Prac_3$ hdfs dfs -ls /
Found 4 items
drwxr-xr-x   - hduser supergroup          0 2025-03-12 23:37 /Media
drwxr-xr-x   - hduser supergroup          0 2025-03-17 23:34 /destination
-rw-r--r--   1 hduser supergroup         46 2025-03-26 23:25 /myfile.txt
drwxr-xr-x   - hduser supergroup          0 2025-03-26 23:40 /output
hduser@paradox:~/BDS_Prac_3$
```

Conclusion: The program on implementation of word count / frequency using MapReduce has been demonstrated successfully.

Practical 4

Aim: Install, configure and run Pig. Execute Pig Latin scripts to sort, group, join, project and filter data.

Start the hadoop and verify all services are started

/usr/local/hadoop/sbin/start-all.sh

```
hduser@paradox:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [paradox]
Starting resourcemanager
Starting nodemanagers
hduser@paradox:~$
hduser@paradox:~$ jps
3889 DataNode
4082 SecondaryNameNode
4852 Jps
3684 NameNode
4486 NodeManager
4360 ResourceManager
```

Download the Pig Package file:

wget https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz

Navigate to /usr/local/

sudo tar xzvf /home/hduser/Downloads/pig-0.17.0.tar.gz

sudo mv pig-0.17.0-src/ pig

Add the Pig environment variables in bashrc and check the pig version to verify the installation.

sudo nano ~/.bashrc

Add the following lines

export PIG_HOME=/usr/local/pig export

PATH=$PATH:$PIG_HOME/bin

export PIG_CLASSPATH=$HADOOP_HOME/conf

```
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin
export PIG_CLASSPATH=$HADOOP_HOME/conf
```

pig –-version

```
hduser@paradox:/usr/local/pig/bin$ pig --version
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for
s.
Apache Pig version 0.17.0 (r1797386)
compiled Jun 02 2017, 15:41:58
hduser@paradox:/usr/local/pig/bin$
```

Create a database file.

sudo nano products.txt

Enter some text like (without spaces)

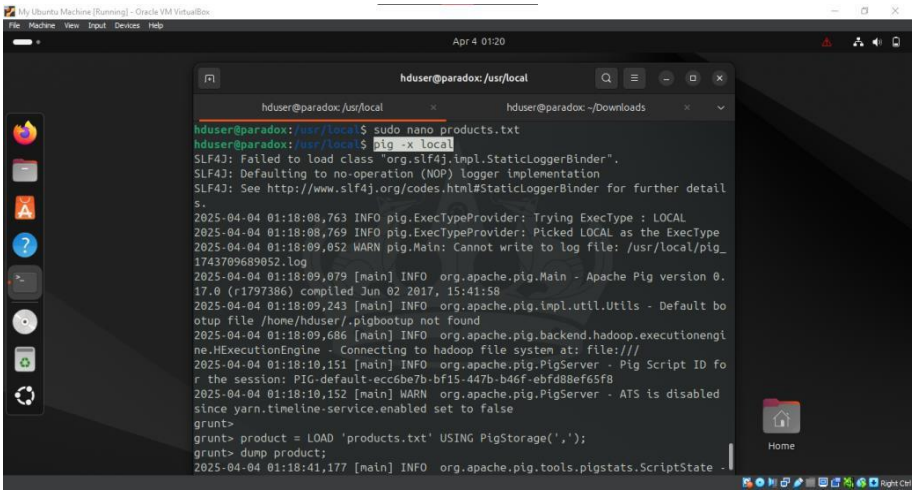
1,phone,45,mumbai,2023

2,laptop,44,pune,2022

```
 GNU nano 7.2                          products.txt
1,phone,45,mumbai,2023
2,laptop,44,pune,2022
```

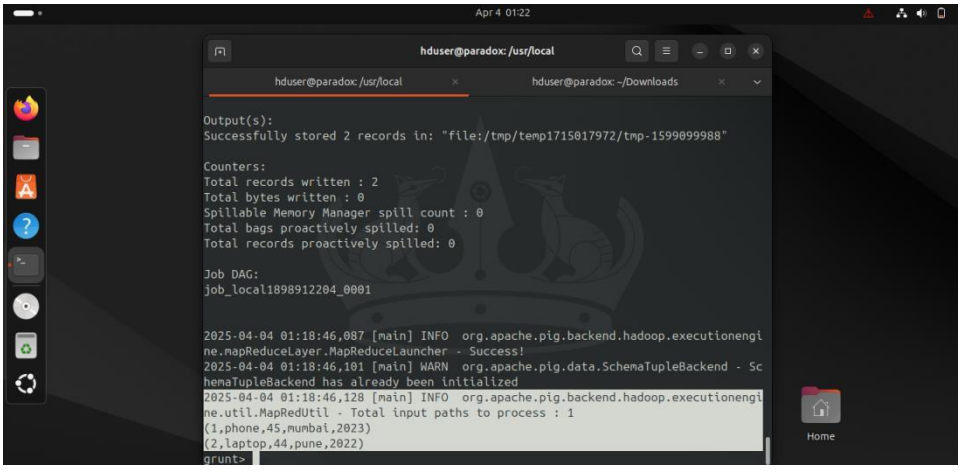
Run the pig in Local mode and load the products file

pig -x local



product = LOAD 'products.txt' USING PigStorage(',');

dump product;



Running in HDFS mode

First we need to move products.txt to HDFS

hdfs dfs -put /usr/local/products.txt /

pig

product = LOAD 'hdfs://localhost:54310/products.txt' USING PigStorage(',');

dump product;

```
grunt>
grunt> product = LOAD 'hdfs://localhost:54310/products.txt' USING PigStorage(','
);
grunt> dump product;
2025-04-04 09:26:40,552 [main] INFO  org.apache.pig.tools.pigstats.ScriptState -
 Pig features used in the script: UNKNOWN
2025-04-04 09:26:40,978 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Ke
y [pig.schematuple] was not set... will not generate code.
2025-04-04 09:26:41,348 [main] INFO  org.apache.pig.newplan.logical.optimizer.Lo
gicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalc
ulator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeF
```

```
Counters:
Total records written : 2
Total bytes written : 5755845
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local193885173_0001


2025-04-04 09:27:05,743 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2025-04-04 09:27:05,822 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2025-04-04 09:27:05,905 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(1,phone,45,mumbai,2023)
(2,laptop,44,pune,2022)
```

Use of DISTINCT operator in PIG. Assume appropriate data in text files.

sudo nano m3.txt

```
  GNU nano 7.2
4,2,5
1,4,1
9,4,5
3,4,7
2,5,6
3,5,9
```

```
grunt> m3 = LOAD 'm3.txt' USING PigStorage(',');
grunt> dump m3;
2025-04-04 10:03:17,891 [main] INFO  org.apache.pig.tools.pigs
```

Use of FILTER operator in PIG

m3 = LOAD 'm3.txt' USING PigStorage(',') as (a1:int,a2:int,a3:int);

result_f = filter m3 by a3==6;

```
grunt> m3 = LOAD 'm3.txt' USING PigStorage(',') as (a1:int,a2:int,a3:int);
grunt> result_f = filter m3 by a3==6;
grunt>
```

dump result_f

```
2025-04-04 10:16:21,827 [main] INFO  org.apache.pig.backe
Launcher - Success!
2025-04-04 10:16:21,830 [main] WARN  org.apache.pig.data.
y been initialized
2025-04-04 10:16:21,843 [main] INFO  org.apache.pig.backe
input paths to process : 1
(2,5,6)
grunt>
```

Use of ORDERBY operator in PIG

result_ob = ORDER m3 BY a1 ASC;

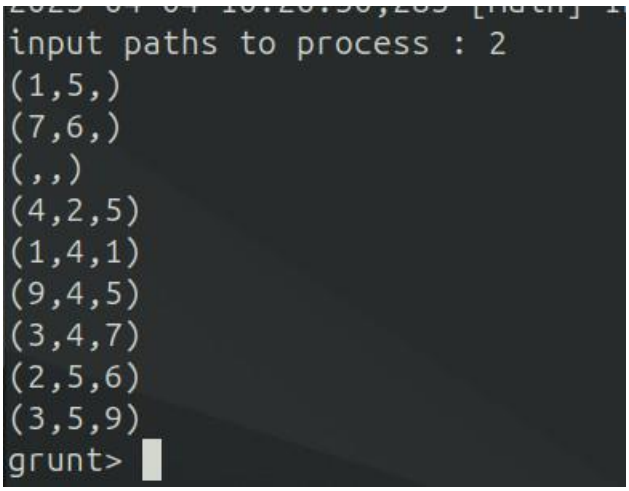
dump result_ob;

```
Launcher - Success!
2025-04-04 10:19:24,825 [main] WARN
y been initialized
2025-04-04 10:19:24,843 [main] INFO
input paths to process : 1
(1,4,1)
(2,5,6)
(3,5,9)
(3,4,7)
(4,2,5)
(9,4,5)
grunt>
```

Use of UNION operator in PIG

Sudo nano m1.txt

```
  GNU nano 7.2
1,5
7,6
```



Conclusion: Practical to Install, configure and run Pig. Execute Pig Latin scripts to sort, group, join, project and filter data, successfully executed.

Practical 5

Aim: Install, configure and run Hive.

Start the hadoop and verify all services are started

/usr/local/hadoop/sbin/start-all.sh

```
hduser@paradox:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [paradox]
Starting resourcemanager
Starting nodemanagers
hduser@paradox:~$
hduser@paradox:~$ jps
3889 DataNode
4082 SecondaryNameNode
4852 Jps
3684 NameNode
4486 NodeManager
4360 ResourceManager
```

Navigate to /usr/local/

sudo tar xvzf /home/hduser/Downloads/apache-hive-3.1.2-bin.tar.gz

sudo mv apache-hive-3.1.2-bin/ hive

sudo chmod 777 hive

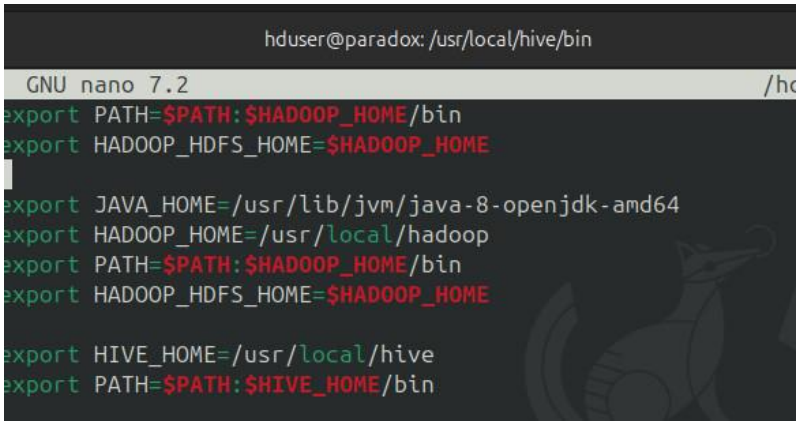
Add the HIVE_HOME path in the bashrc file.

sudo nano ~/.bashrc

#add following lines

export HIVE_HOME=/usr/local/hive

export PATH=$PATH:$HIVE_HOME/bin



source ~/.bashrc

Change the directory to /usr/local/hive/bin

and add the following lines in hive-config.sh

cd /usr/local/hive/bin

sudo nano hive-config.sh

export HADOOP_HOME=/usr/local/hadoop

```
# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}

export HADOOP_HOME=/usr/local/hadoop
```

Hive is installed now, but we need to first create some directories in HDFS for Hive to store its data

hdfs dfs -mkdir /tmp

hdfs dfs -chmod g+w /tmp

hdfs dfs -chmod o+w /tmp

hdfs dfs -mkdir -p /user/hive/warehouse

hdfs dfs -chmod g+w /user/hive/warehouse

hdfs dfs -chmod 777 /tmp

hdfs dfs -chown -R hduser:supergroup /user/hive/warehouse

hdfs dfs -chmod -R 777 /user/hive/warehouse

hdfs dfs -ls /

```
hduser@paradox:/usr/local/hive/bin$ hdfs dfs -ls /
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Found 6 items
drwxr-xr-x   - hduser supergroup          0 2025-03-12 23:37 /Media
drwxr-xr-x   - hduser supergroup          0 2025-03-17 23:34 /destination
-rw-r--r--   1 hduser supergroup         46 2025-03-26 23:25 /myfile.txt
drwxr-xr-x   - hduser supergroup          0 2025-03-26 23:40 /output
drwxrwxrwx   - hduser supergroup          0 2025-04-03 18:45 /tmp
drwxr-xr-x   - hduser supergroup          0 2025-04-03 09:59 /user
```

Initialze the database schema

cd /$HIVE_HOME/bin

sudo ./schematool -initSchema -dbType derby

There is compatibility error between Hadoop and Hive guava versions.

To fix the NoSuchMethodError, Locate the guava jar file in the Hive lib directory

Remove the guava jar file from /hive/lib

sudo rm lib/guava-19.0.jar

Copy the guava jar from hadoop lib to hive lib directory

sudo cp $HADOOP_HOME/share/hadoop/common/lib/guava-27.0-jre.jar /usr/local/hive/lib/

Once copied, Use the schematool command once again to initiate the Derby database.

cd /$HIVE_HOME/bin

sudo ./schematool -initSchema -dbType derby

Note for Error: FUNCTION 'NUCLEUS_ASCII' -

sudo rm -rf metastore_db

Start HIVE:

cd /usr/local/hive/bin

sudo ./hive

```
hduser@paradox:/usr/local/hive/bin$ sudo ./hive
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further
Hive Session ID = 470c2d06-922c-4727-ad6f-9248b6633372

Logging initialized using configuration in jar:file:/usr/local/hive/lib/H
e
Hive-on-MR is deprecated in Hive 2 and may not be available in the future
(i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 86b6bbdc-c789-4393-9858-73c061dcb59e
```

Create a database and show

create database company

```
Hive Session ID = 86b6bbdc-c789-4393-9858-73c061dcb59e
hive> create database company;
OK
Time taken: 3.251 seconds
hive> show databases;
OK
company
default
Time taken: 6.16 seconds, Fetched: 2 row(s)
```

Create Employee table

create table employees (id int, name string, country string, department string, salary int) row format delimited fields terminated by ' ';

```
hive> create table employees (id int, name string, country string, department string, salary int) row format delimited fields te
rminated by ' ';
OK
Time taken: 14.864 seconds
hive>
    > show tables;
OK
employees
Time taken: 1.687 seconds, Fetched: 1 row(s)
```

Load the data into a table from a file

sudo nano employees.txt

Enter few rows without spaces

```
hduser@paradox: /usr/local/hive/bin
 GNU nano 7.2
1 Furkan India ComputerScience 65700
2 John Armenia IT 45679
3 Test Mexci CS 35678
```

load data local inpath "./employees.txt" into table employees;

```
hive>
hive> load data local inpath "./employees.txt" into table employees;
Loading data to table default.employees
OK
Time taken: 2.688 seconds
hive>
```

Reading the Table Data

select * from employees;

```
hive> select * from employees;
OK
NULL    faculty udit    45000   NULL
NULL    industry        l&t     80000   NULL
NULL    Cyber   SSD     87655   NULL
1       Furkan  India   ComputerScience 65700
2       John    Armenia IT      45679
3       Test    Mexci   CS      35678
Time taken: 2.881 seconds, Fetched: 6 row(s)
```

Conclusion: The Practical to install, configure and execute hive is successfully executed.

Practical 6

Aim: Implement Bucketing using Hive

Start the hadoop and verify all services are started

/usr/local/hadoop/sbin/start-all.sh

```
hduser@paradox:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [paradox]
Starting resourcemanager
Starting nodemanagers
hduser@paradox:~$
hduser@paradox:~$ jps
3889 DataNode
4082 SecondaryNameNode
4852 Jps
3684 NameNode
4486 NodeManager
4360 ResourceManager
```

Start Hive Create a database called "show_bucket" , Create a table named "emp_demo" in show_bucket.db. Assume appropriate columns

sudo ./hive
create database show_bucket;

```
hive> create database show_bucket;
OK
Time taken: 1.572 seconds
hive>
```

use show_bucket;

create table emp_demo (id int, name string, salary int)
row format delimited
fields terminated by ',';

```
hive> create table emp_demo (id int, name string, salary int)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 2.279 seconds
hive>
    > show tables;
OK
emp_demo
Time taken: 0.718 seconds, Fetched: 1 row(s)
hive>
```

Create emp_details.txt, assume appropriate data. Load data in emp_demo table from file emp_details.txt.

```
hduser@paradox: ~   hduser@paradox: /usr/loc...   hd
  GNU nano 7.2                    emp_details.txt
1,Furkan,85600
2,Abdullah,45000
3,Arman,76000
4,Sahil,75444
5,Avaish,85444
```

load data local inpath 'emp_details.txt' into table emp_demo;

```
hive> load data local inpath 'emp_details.txt' into table emp_demo;
Loading data to table show_bucket.emp_demo
OK
Time taken: 12.484 seconds
hive>
```

Verify the employee table along with its schema from terminal as well as browser .

```
hive> DESCRIBE emp_demo;
OK
id                      int
name                    string
salary                  int
Time taken: 5.69 seconds, Fetched: 3 row(s)
```

Browse Directory

/user/hive/warehouse/show_bucket.db/emp_demo Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	75 B	Apr 06 12:32	1	128 MB	emp_details.txt

Showing 1 to 1 of 1 entries Previous 1 Next

Hadoop, 2022.

Enable the bucketing, Create a bucketing table "emp_bucket"

set hive.enforce.bucketing = true;

create table emp_bucket (id int, name string, salary int)

> row format delimited

> fields terminated by ',';

```
hive> set hive.enforce.bucketing = true;
hive> create table emp_bucket (id int, name string, salary int)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.177 seconds
hive>
```

Insert the data of emp_demo table into the bucketed table.

insert overwrite table emp_bucket select * from emp_demo;

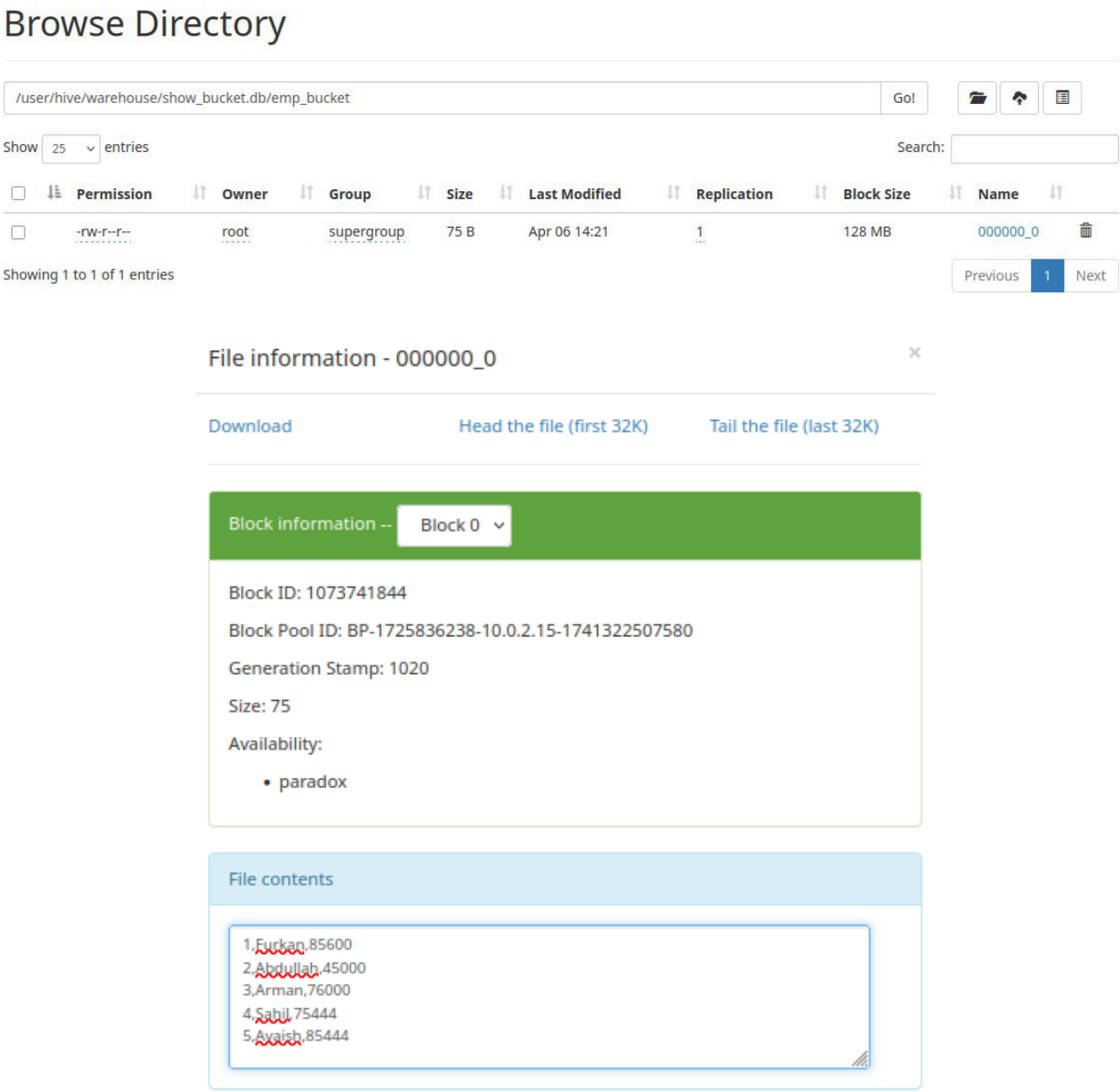
```
hive> insert overwrite table emp_bucket select * from emp_demo;
Query ID = root_20250406142112_b88b9fdd-f4b6-4e5c-85f4-a3522b503eb9
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<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:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-04-06 14:21:34,695 Stage-1 map = 0%,  reduce = 0%
2025-04-06 14:21:36,869 Stage-1 map = 100%,  reduce = 0%
2025-04-06 14:21:38,385 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local385817976_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:54310/user/hive/warehouse/show_bucket
.db/emp_bucket/.hive-staging_hive_2025-04-06_14-21-12_594_1757665977390065758-1
/-ext-10000
```

```
Loading data to table show_bucket.emp_bucket
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 150 HDFS Write: 456 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 28.286 seconds
hive>
```

Verify the output from terminal and browser

hdfs dfs -ls /user/hive/warehouse/show_bucket.db/emp_bucket

```
hduser@paradox:~$ hdfs dfs -ls /user/hive/warehouse/show_bucket.db/emp_bucket
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further detai
ls.
Found 1 items
-rw-r--r--   1 root supergroup         75 2025-04-06 14:21 /user/hive/warehouse
/show_bucket.db/emp_bucket/000000_0
hduser@paradox:~$
```



Conclusion: The practical for implementing bucketing in Hive was successfully completed.

Practical 7

Aim: Install, configure and run Apache Spark. Create & transform RDDs

Install Scala:

sudo apt install scala -y

```
hduser@paradox:~$ scala -version
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
hduser@paradox:~$
```

Download and Install Apache Spark in path /opt

sudo wget https://downloads.apache.org/spark/spark-3.5.5/spark-3.5.5-bin-hadoop3.tgz

```
hduser@paradox:/opt$ sudo wget https://downloads.apache.org/spark/spark-3.5.5/s
park-3.5.5-bin-hadoop3.tgz
--2025-04-06 15:53:22--  https://downloads.apache.org/spark/spark-3.5.5/spark-3
.5.5-bin-hadoop3.tgz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181
.214.104, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443...
 connected.
HTTP request sent, awaiting response... 200 OK
Length: 400724056 (382M) [application/x-gzip]
Saving to: 'spark-3.5.5-bin-hadoop3.tgz'

spark-3.5.5-bin-had 100%[===================>] 382.16M   625KB/s    in 18m 19s

2025-04-06 16:13:56 (356 KB/s) - 'spark-3.5.5-bin-hadoop3.tgz' saved [400724056
/400724056]
```

Extract the archive and rename the directory.

```
hduser@paradox:/opt$ sudo mv spark-3.5.5-bin-hadoop3 spark
hduser@paradox:/opt$
```

Set Up Environment Variables

Add the following lines at the end of .bashrc:

export SPARK_HOME=/opt/spark

export PATH=$SPARK_HOME/bin:$SPARK_HOME/sbin:$PATH

export PYSPARK_PYTHON=/usr/bin/python3

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

Start Spark Master

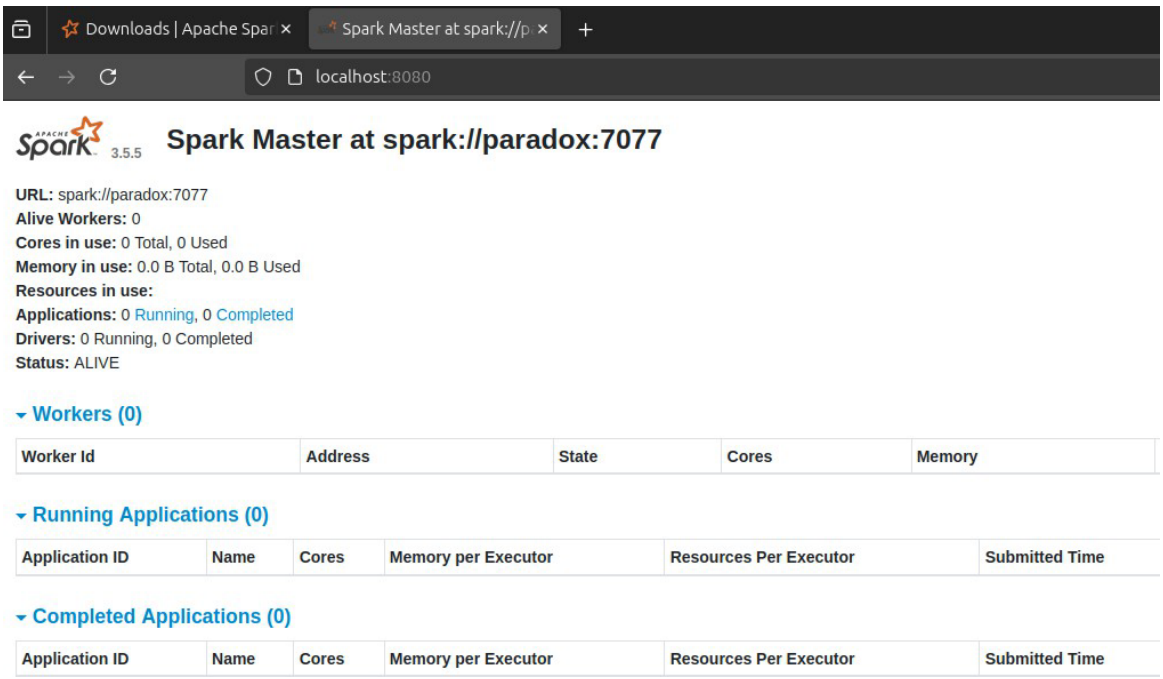
sudo /opt/spark/sbin/start-master.sh

You can now access the **Spark Master Web UI** to confirm everything is working:

Open your browser and go to:

http://localhost:8080

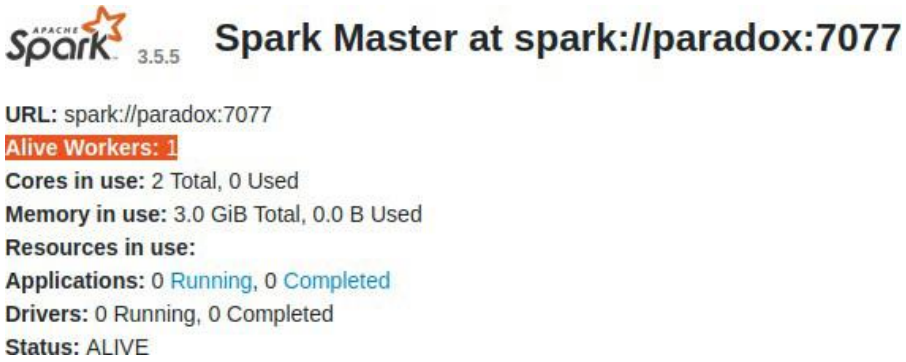
```
hduser@paradox:~$ sudo /opt/spark/sbin/start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spar
k-root-org.apache.spark.deploy.master.Master-1-paradox.out
hduser@paradox:~$
```



Start Spark Worker

start-worker.sh spark://yourhostname:7077

Now go to localhost:8080 Active Worker is set to one



Start Spark Shell

spark-shell

Verify the installation by running:

sc.appName

```
scala> sc.appName
res0: String = Spark shell
```

For Creating and Transforming RDDs

Launch PySpark

```
Using Python version 3.12.3 (main, Feb  4 2025 14:48:35)
Spark context Web UI available at http://paradox:4040
Spark context available as 'sc' (master = local[*], app id = local-17439380
7).
SparkSession available as 'spark'.
>>>
```

The .collect() Action:

```
>>> rdd = sc.parallelize([1, 2, 3, 4, 5])
>>> print(rdd.collect())
[Stage 0:>
[Stage 0:>

[1, 2, 3, 4, 5]
```

The .count() Action:

```
>>> count_rdd = sc.parallelize([1, 2, 3, 4, 5,6,7,9])
>>> count_rdd.count()
[Stage 1:>
[Stage 1:============================>

8
```

The .first() Action

```
>>> first_rdd = sc.parallelize([1, 2, 3, 4, 5,6,7,9])
>>> first_rdd.first()
[Stage 2:>

1
```

The .take() Action

```
>>> first_rdd.take(4)
[Stage 3:>

[1, 2, 3, 4]
```

The .reduce() Action

```
>>> rdd.reduce(lambda x,y:x+y)
[Stage 4:>
[Stage 4:============================>

15
```

The saveAsTextFile() Action

```
>>> save_rdd.saveAsTextFile('rdd.txt')
[Stage 5:>
```

The .map() transformation

```
>>> rdd.collect()
[1, 2, 3, 4, 5]
>>> rdd.map(lambda x:x+10).collect()
[11, 12, 13, 14, 15]
>>>
```

The .filter() transformation

```
>>> rdd.collect()
[1, 2, 3, 4, 5]
>>> rdd.filter(lambda x:x%2==0).collect()
[2, 4]
>>>
```

The Union Transformation

```
>>> my_rdd = sc.parallelize([1,2,3,4,5,6,7,8,9,10])
>>> u_rdd_1 = my_rdd.filter(lambda x:x%2==0)
>>> u_rdd_2 = my_rdd.filter(lambda x:x%3==0)
>>> u_rdd_1.un
u_rdd_1.union(      u_rdd_1.unpersist(
>>> u_rdd_1.union(u_rdd_2).collect()
[2, 4, 6, 8, 10, 3, 6, 9]
>>>
```

The FlatMap Transformation

```
>>> flat_map = sc.parallelize(["Hey there","I am Furkan"])
>>> flat_map.flatMap(lambda x:x.split(" ")).collect()
['Hey', 'there', 'I', 'am', 'Furkan']
>>>
```

Conclusion: The practical to implement actions & transformation on RDDS using apache spark is successfully completed.

Practical 8

Aim: Write a program to implement an application that stores big data in Hbase/ MongoDB & manipulate it using R/Python.

Code:

Step 1: Install mongodb by executing the installation file "mongodb-windows-x86_64-4.4.6-signed"

Click next, next and finish the installation

Step2: Launch MongoDB

Navigate to the following location: "C:\Program Files\MongoDB\Server\4.4\bin"

#Start mongo daemon -

mongod

#Start mongo service -

mongosh

 # Creating Collections and Documents

A MongoDB database is a physical container for collections of documents. Each database gets its own set of files on the file system. These files are managed by the MongoDB server, which can handle several databases.

In MongoDB, a collection is a group of documents. Collections are somewhat analogous to tables in a traditional RDBMS, but without imposing a rigid schema. In theory, each document in a collection can have a completely different structure or set of fields.

 # Show list of db

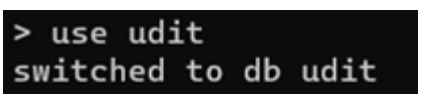
show dbs

 # Show current db

db

 # Create/switch to a db - use dbname

use udit



 # Display existing collections

show collections;

```
> show collections;
subjects
```

 # Create collection - db.collectionname

db.subjects

```
> db.subjects
udit.subjects
```

 # Insert into collection

db.subjects.insertOne({"name":"bda"})
db.subjects.insertOne({"name":"mn"})
db.subjects.insertOne({"name":"ip"})
db.subjects.insertOne({"name":"msa"})
show collections;

```
> db.subjects.insert({"name":"bda"})
WriteResult({ "nInserted" : 1 })
> db.subjects.insert({"name":"mn"})
WriteResult({ "nInserted" : 1 })
> db.subjects.insert({"name":"ip"})
WriteResult({ "nInserted" : 1 })
> db.subjects.insert({"name":"msa"})
WriteResult({ "nInserted" : 1 })
> show collections;
subjects
```

 # Display all records in collection

db.subjects.find();

```
> db.subjects.find();
{ "_id" : ObjectId("649938752c79768509eaa91a"), "name" : "bda" }
{ "_id" : ObjectId("649938752c79768509eaa91b"), "name" : "mn" }
{ "_id" : ObjectId("649938752c79768509eaa91c"), "name" : "ip" }
{ "_id" : ObjectId("649938752c79768509eaa91d"), "name" : "msa" }
>
```

 # Display specific record in colleciton

db.subjects.find({"name": "bda"});

```
> db.subjects.find({"name": "bda"});
{ "_id" : ObjectId("649938752c79768509eaa91a"), "name" : "bda" }
>
```

 # Using MongoDB With Python and PyMongo
 # Install python 3.7.4
 # Launch IDLE 3.7

 # Installing PyMongo (in cmd)
 # MongoDB provides an official Python driver called PyMongo.

python -m pip install pymongo

#
```
C:\Users\Gulzarina>pip install pymongo
Collecting pymongo
  Downloading pymongo-4.4.0-cp311-cp311-win_amd64.whl (453 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 453.6/453.6 kB 359.3 kB/s eta 0:00:00
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 283.7/283.7 kB 530.9 kB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.3.0 pymongo-4.4.0
```

Start a Python interactive session and run the following import:

```
import pymongo
```

#Working With Databases, Collections

 # Program 1: Creating a Database

```
from pymongo import MongoClient          //import MongoClient from pymongo.
```

Create a client object to communicate with running MongoDB instance

```
myclient = MongoClient()
myclient                                                    //test client
```

```
>>> import pymongo
...
>>> from pymongo import MongoClient
>>> myclient = MongoClient()
>>> myclient
MongoClient(host=['localhost:27017'], document_class=dict,
tz_aware=False, connect=True)
>>>
```

To provide a custom host and port when you need to provide a host and port that differ from MongoDB's default

```
myclient = MongoClient(host="localhost", port=27017)
```

```
>>> myclient = MongoClient(host="localhost", port=27017)
```

Check db list

```
print(myclient.list_database_names())
```

```
>>> print(myclient.list_database_names())
['admin', 'config', 'local', 'mlib', 'udit']
```

Define which database you want to use

```
db = myclient["udit"]
```

Progam 2: Creating a Collection

```
import pymongo
myclient = MongoClient(host="localhost", port=27017)
db = myclient.mlib
col=db.subjects1                                    /// create collection
dict = {"name":"ds", "sem":"1"}                     /// create dictionary
```

x=col.insert_one(dict) */// insert into collection*

```
import pymongo
myclient = MongoClient(host="localhost", port=27017)
db = myclient.mlib
col=db.subjects1
dict = {"name":"ds", "sem":"1"}
x=col.insert_one(dict)
```

print(client1.list_database_names())

```
print(myclient.list_database_names())
['admin', 'config', 'local', 'mlib', 'udit']
```

Conclusion: Program performed to implement an application that stores big data in MongoDB & manipulate it using Python has been demonstrated successfully.

Practical 9

Aim: Install, configure and run Apache Strom

Install ZooKeeper

sudo apt install -y zookeeperd

Start the ZooKeeper service:

sudo systemctl start zookeeper
sudo systemctl enable zookeeper
sudo systemctl status zookeeper

```
hduser@paradox:/opt$ sudo systemctl status zookeeper
[sudo] password for hduser:
● zookeeper.service - LSB: centralized coordination service
     Loaded: loaded (/etc/init.d/zookeeper; generated)
     Active: active (running) since Sun 2025-04-06 22:56:46 IST; 58min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 6039 ExecStart=/etc/init.d/zookeeper start (code=exited, status=0/
      Tasks: 29 (limit: 4784)
     Memory: 48.2M (peak: 48.4M)
        CPU: 48.316s
     CGroup: /system.slice/zookeeper.service
             └─6050 /usr/bin/java -cp /etc/zookeeper/conf:/usr/share/java/zooke

Apr 06 22:56:46 paradox systemd[1]: Starting zookeeper.service - LSB: centraliz
Apr 06 22:56:46 paradox systemd[1]: Started zookeeper.service - LSB: centralize
```

```
hduser@paradox:/opt$ echo "ruok" | nc localhost 2181
imokhduser@paradox:/opt$
```

Download and Install Apache Storm in /opt

sudo wget https://dlcdn.apache.org/storm/apache-storm-2.8.0/apache-storm-2.8.0.tar.gz

```
hduser@paradox:/opt$ sudo wget https://dlcdn.apache.org/storm/apache-storm-2.8.0
/apache-storm-2.8.0.tar.gz
--2025-04-06 23:23:07--  https://dlcdn.apache.org/storm/apache-storm-2.8.0/apach
e-storm-2.8.0.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 345052924 (329M) [application/x-gzip]
Saving to: 'apache-storm-2.8.0.tar.gz'

apache-storm-2.8.0. 100%[===================>] 329.07M   488KB/s    in 15m 13s
```

Extract the archive and rename it to Storm:

```
hduser@paradox:/opt$ sudo mv apache-storm-2.8.0 storm
hduser@paradox:/opt$ ls
apache-storm-2.8.0.tar.gz  spark-3.5.5-bin-hadoop3.tgz
solr                       storm
solr-9.4.1                 VBoxGuestAdditions-7.0.22
spark
```

Configure Apache Storm

Add this lines in nano /opt/storm/conf/storm.yaml

```
storm.zookeeper.servers:
        - "localhost"

        nimbus.seeds: ["localhost"]

        supervisor.slots.ports:
  - 6700
  - 6701
  - 6702
  - 6703
ui.port: 8080
```

Create the local directory:

sudo mkdir /opt/storm/tmp

Start Apache Storm Services

In separate terminal tabs or with tmux/screen, run the following components:

Nimbus (Master node):

cd /opt/storm

sudo bin/storm nimbus

Supervisor (Worker node):

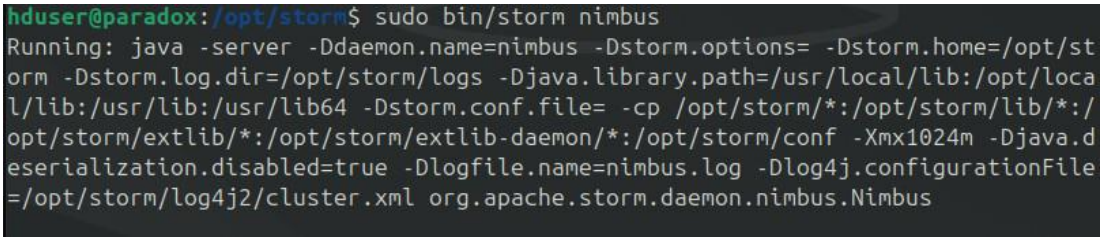
cd /opt/storm

sudo bin/storm supervisor

Storm UI (Web interface):

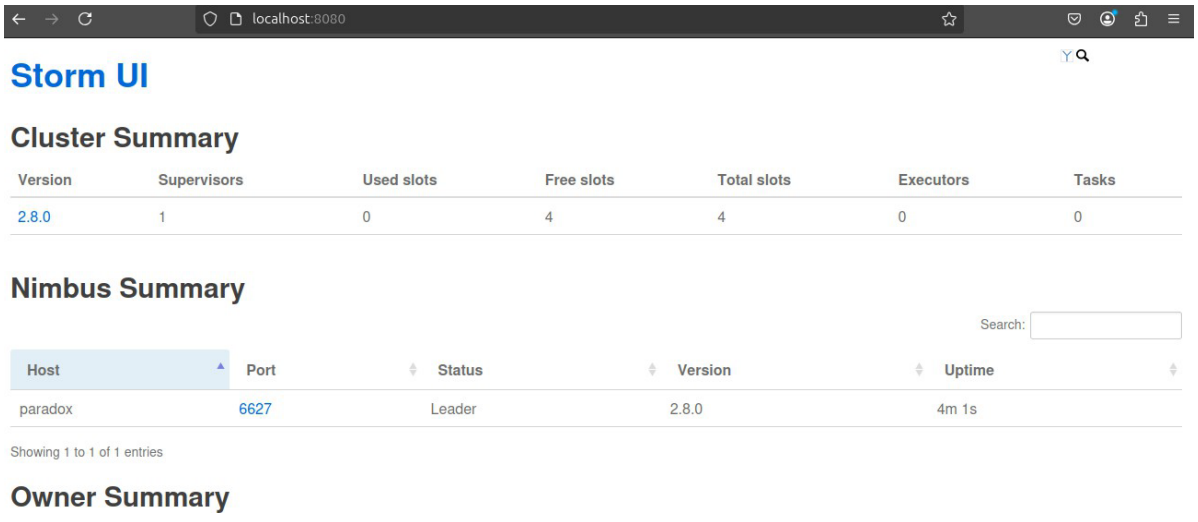
cd /opt/storm

sudo bin/storm ui



```
hduser@paradox:/opt/storm$ sudo bin/storm supervisor
[sudo] password for hduser:
Running: java -server -Ddaemon.name=supervisor -Dstorm.options= -Dstorm.home=/opt/storm -Dstorm.log.dir=/opt/storm/logs -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/lib64 -Dstorm.conf.file= -cp /opt/storm/*:/opt/storm/lib/*:/opt/storm/extlib/*:/opt/storm/extlib-daemon/*:/opt/storm/conf -Xmx256m -Djava.deserialization.disabled=true -Dlogfile.name=supervisor.log -Dlog4j.configurationFile=/opt/storm/log4j2/cluster.xml org.apache.storm.daemon.supervisor.Supervisor
```

```
hduser@paradox:/opt/storm$ sudo bin/storm ui
Running: java -server -Ddaemon.name=ui -Dstorm.options= -Dstorm.home=/opt/storm -Dstorm.log.dir=/opt/storm/logs -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/lib64 -Dstorm.conf.file= -cp /opt/storm/*:/opt/storm/lib/*:/opt/storm/extlib/*:/opt/storm/extlib-daemon/*:/opt/storm/lib-webapp/*:/opt/storm/conf -Xmx768m -Djava.deserialization.disabled=true -Dlogfile.name=ui.log -Dlog4j.configurationFile=/opt/storm/log4j2/cluster.xml org.apache.storm.daemon.ui.UIServer
Apr 07, 2025 12:42:12 AM org.glassfish.jersey.message.internal.MessagingBinders$EnabledProvidersBinder bindToBinder
WARNING: A class jakarta.activation.DataSource for a default provider MessageBodyWriter<jakarta.activation.DataSource> was not found. The provider is not available.
Apr 07, 2025 12:42:12 AM org.glassfish.jersey.server.wadl.WadlFeature configure
WARNING: JAX-B API not found . WADL feature is disabled.
```



Download the SampleTopology

sudo wget https://repo1.maven.org/maven2/org/apache/storm/storm-starter/2.8.0/storm-starter-2.8.0.jar

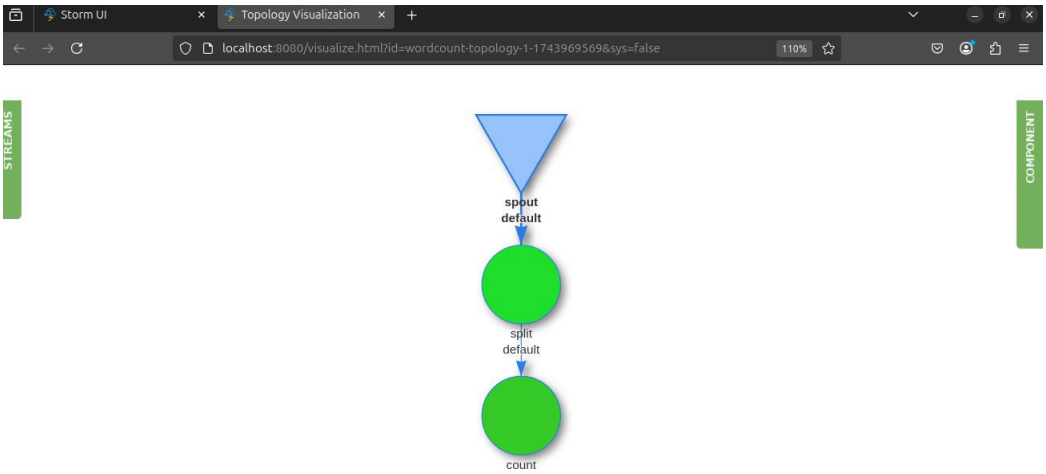
```
hduser@paradox:/opt/storm$ sudo wget https://repo1.maven.org/maven2/org/apache/storm/storm-starter/2.8.0/storm-starter-2.8.0.jar
--2025-04-07 01:10:13--  https://repo1.maven.org/maven2/org/apache/storm/storm-starter/2.8.0/storm-starter-2.8.0.jar
Resolving repo1.maven.org (repo1.maven.org)... 199.232.196.209, 199.232.192.209, 2a04:4e42:4d::209, ...
```

Submit the Topology

```
hduser@paradox:/opt/storm$ bin/storm jar storm-starter-2.8.0.jar org.apache.stor
m.starter.WordCountTopology wordcount-topology
Running: /usr/lib/jvm/java-17-openjdk-amd64/bin/java -client -Ddaemon.name= -Dst
orm.options= -Dstorm.home=/opt/storm -Dstorm.log.dir=/opt/storm/logs -Djava.libr
ary.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/lib64 -Dstorm.conf.file= -c
p /opt/storm/*:/opt/storm/lib-worker/*:/opt/storm/extlib/*:storm-starter-2.8.0.j
ar:/opt/storm/conf:/opt/storm/bin: -Dstorm.jar=storm-starter-2.8.0.jar -Dstorm.d
ependency.jars= -Dstorm.dependency.artifacts={} org.apache.storm.starter.WordCou
ntTopology wordcount-topology
01:29:25.526 [main] INFO  o.a.s.StormSubmitter - Generated ZooKeeper secret payl
oad for MD5-digest: -9095026209636248440:-7669095391788071585
01:29:26.079 [main] INFO  o.a.s.u.NimbusClient - Found leader nimbus : paradox:6
627:0
01:29:26.090 [main] INFO  o.a.s.s.a.ClientAuthUtils - Got AutoCreds []
01:29:26.310 [main] INFO  o.a.s.StormSubmitter - Uploading dependencies - jars..
.
01:29:26.325 [main] INFO  o.a.s.StormSubmitter - Uploading dependencies - artifa
cts...
01:29:26.333 [main] INFO  o.a.s.StormSubmitter - Dependency Blob keys - jars : [
] / artifacts : []
```

```
01:29:28.534 [main] INFO  o.a.s.StormSubmitter - Submitting topology wordcount-t
opology in distributed mode with conf {"storm.zookeeper.topology.auth.scheme":"d
igest","storm.zookeeper.topology.auth.payload":"-9095026209636248440:-7669095391
788071585","topology.workers":3,"topology.debug":true}
01:29:31.423 [main] INFO  o.a.s.StormSubmitter - Finished submitting topology: w
ordcount-topology
hduser@paradox:/opt/storm$
```

Topology Visualization



Stop the Topology

bin/storm kill wordcount-topology

```
hduser@paradox:/opt/storm$ bin/storm kill wordcount-topology -w 5

Running: /usr/lib/jvm/java-17-openjdk-amd64/bin/java -client -Ddaemon.name= -Dst
```

Conclusion: Practical to Install, configure and run Apache Strom is successfully completed.

Practical 10

Aim: Install, configure and run Apache Solr.

Download the Solr zip file:

wget https://archive.apache.org/dist/solr/solr/9.4.1/solr-9.4.1.tgz

```
hduser@paradox:~/Desktop$ wget https://archive.apache.org/dist/solr/solr/9.4.1/
solr-9.4.1.tgz
--2025-04-06 16:41:30--  https://archive.apache.org/dist/solr/solr/9.4.1/solr-9
.4.1.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1
a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 280852760 (268M) [application/x-gzip]
Saving to: ‘solr-9.4.1.tgz’

solr-9.4.1.tgz      100%[==================>] 267.84M   249KB/s    in 16m 18s

2025-04-06 16:57:49 (280 KB/s) - ‘solr-9.4.1.tgz’ saved [280852760/280852760]
```

Extract and Install

tar xzf solr-9.4.1.tgz

cd solr-9.4.1/

To install Solr as a system service:
sudo ./install_solr_service.sh /home/hduser/Desktop/solr-9.4.1.tgz

```
Service solr installed.
Customize Solr startup configuration in /etc/default/solr.in.sh
● solr.service - LSB: Controls Apache Solr as a Service
     Loaded: loaded (/etc/init.d/solr; generated)
     Active: active (exited) since Sun 2025-04-06 17:34:35 IST; 5s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 40777 ExecStart=/etc/init.d/solr start (code=exited, status=0/SUC>
        CPU: 29ms

Apr 06 17:34:22 paradox systemd[1]: Starting solr.service - LSB: Controls Apac>
Apr 06 17:34:22 paradox su[40779]: (to solr) root on none
Apr 06 17:34:22 paradox su[40779]: pam_unix(su-l:session): session opened for >
Apr 06 17:34:35 paradox systemd[1]: Started solr.service - LSB: Controls Apach>
lines 1-11/11 (END)
```

After installation, start Solr with:
sudo systemctl start solr

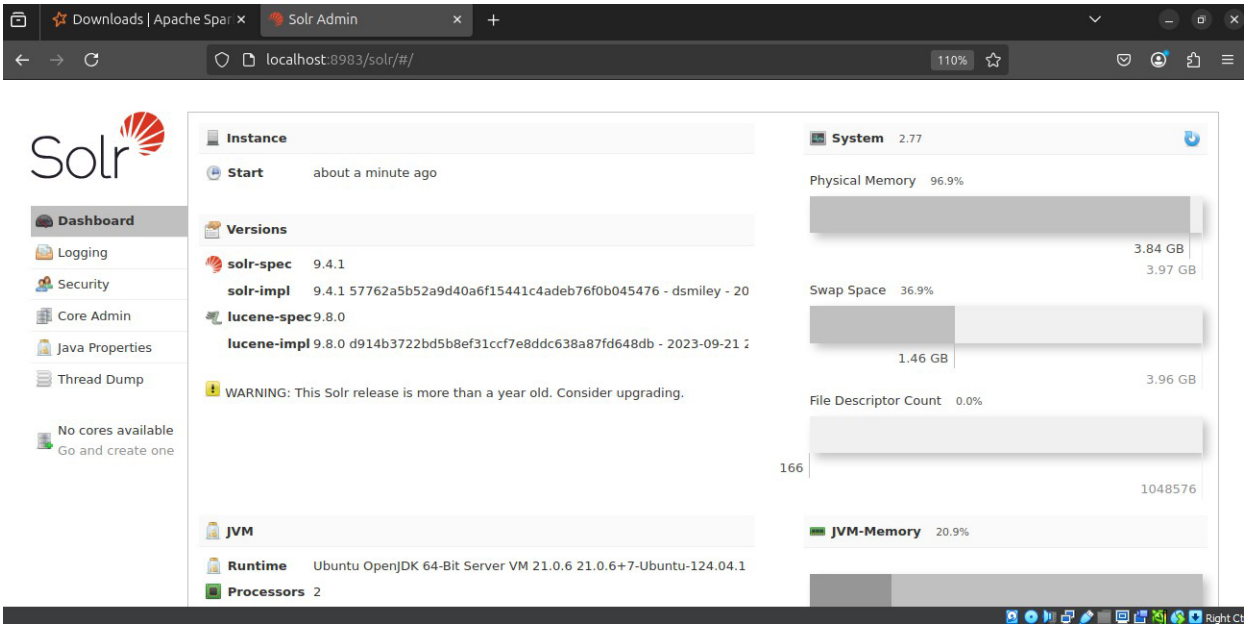
To check if it's running:

sudo systemctl status solr

To stop Solr:

sudo systemctl stop solr

For Solr Dashboard go to http://localhost:8983/solr



Configure Apache Solr

Create a Collection

sudo su – solr

A collection is equivalent to a database in SQL.

```
solr@paradox:~$
solr@paradox:~$ /opt/solr/bin/solr create -c furkan_collection -n _default
WARNING: Using _default configset with data driven schema functionality. NOT RE
COMMENDED for production use.
         To turn off: bin/solr config -c furkan_collection -p 8983 -action set-
user-property -property update.autoCreateFields -value false

Created new core 'furkan_collection'
solr@paradox:~$
```

Change Solr Configuration

Edit Core Config Files

Solr stores configurations in solrconfig.xml and schema.xml inside:

/var/solr/data/mydreams/conf/

1. **Common Settings (Global config)**

   File: /etc/default/solr.in.sh

   Change Port: SOLR_PORT=8984

   Set Memory: SOLR_HEAP="2g"

2. **Enable Basic Authentication**

   Edit /etc/default/solr.in.sh and add:

SOLR_AUTH_TYPE="basic"

SOLR_AUTHENTICATION_OPTS="-Dbasicauth=admin:admin123"

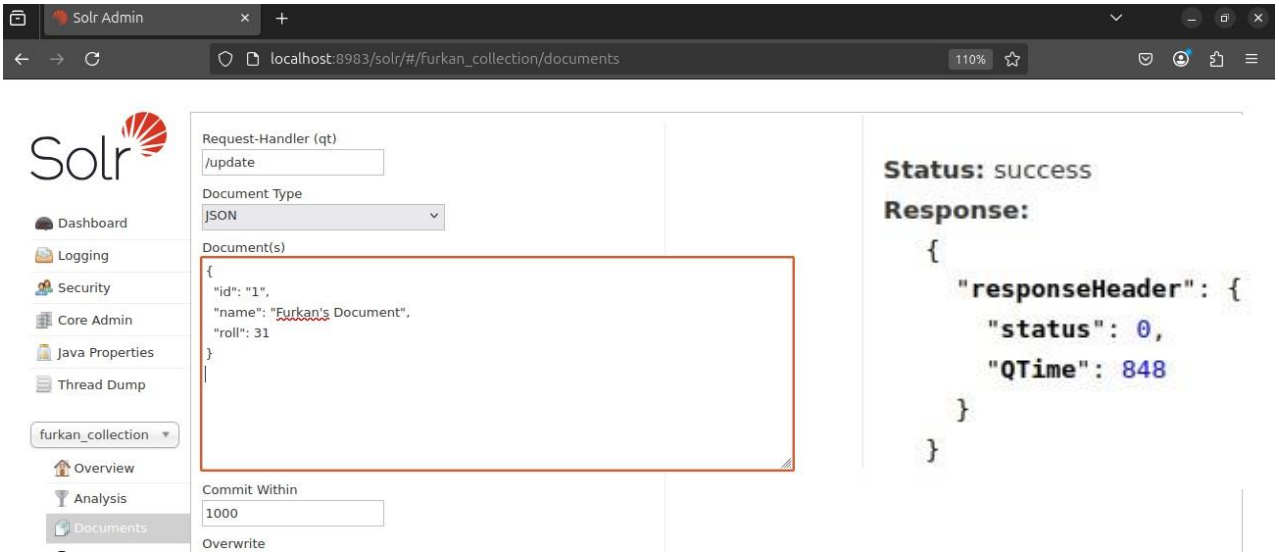
Replace admin:admin123 with your preferred username and password.

Restart Solr to Apply Changes

sudo systemctl restart solr

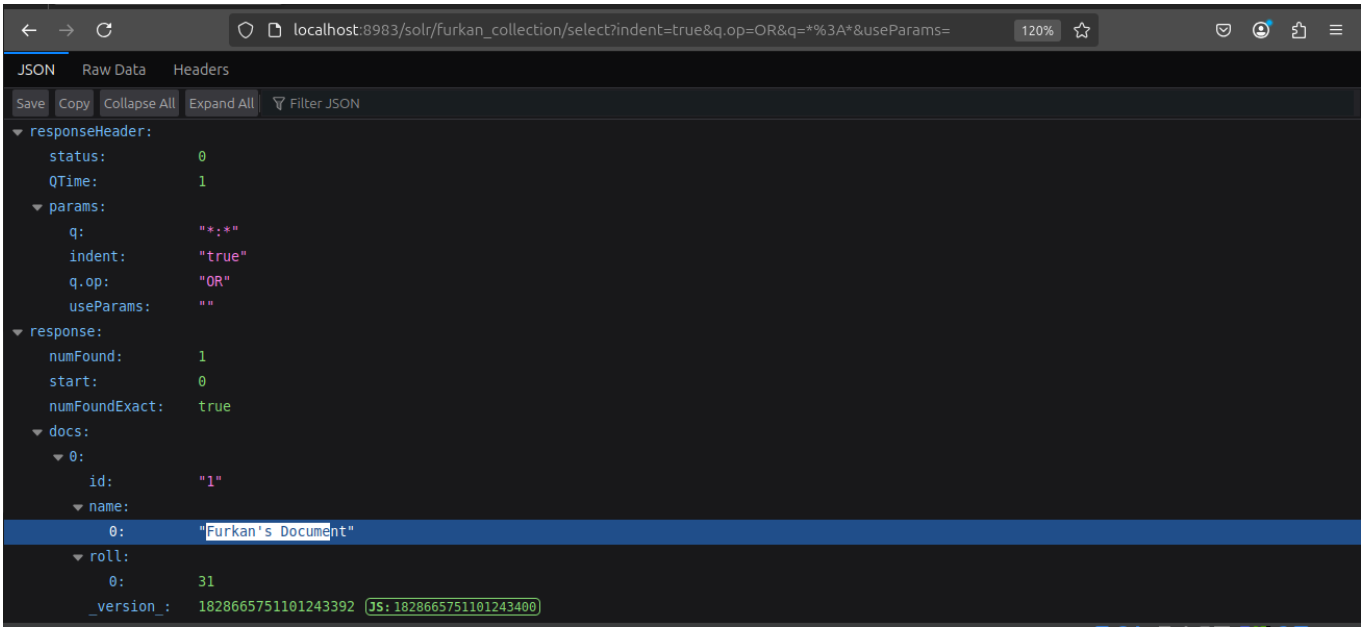
```
SOLR_PID_DIR="/var/solr"
SOLR_HOME="/var/solr/data"
LOG4J_PROPS="/var/solr/log4j2.xml"
SOLR_LOGS_DIR="/var/solr/logs"
SOLR_PORT="8984"
SOLR_HEAP="2g"

SOLR_AUTH_TYPE="basic"
SOLR_AUTHENTICATION_OPTS="-Dbasicauth=furkan:furkan123"
```

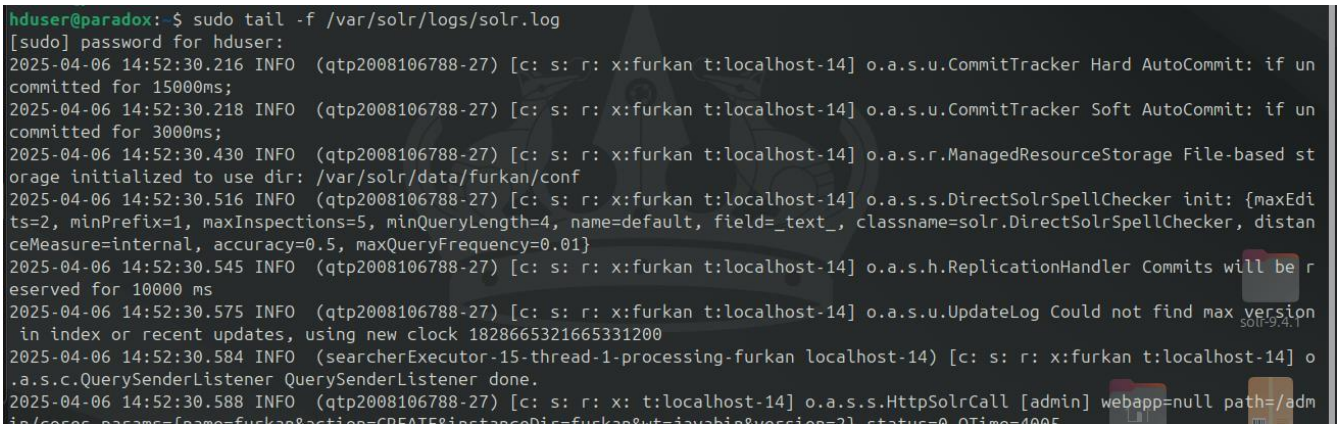
Adding the JSON Documemts



Reading the Documents using CURL:

Reading Document in Browser:



Tail Live Logs

tail -f /var/solr/logs/solr.log