NAME: NEEMA NDANU **ID NO: 666457**
COURSE NO : DSA 4010
COURSE : BIG DATA ANALYTICS

HADOOP - SQL - SQOOP - HDFS PIPELINE

STEP 1: DOWNLOAD AND CONFIGURE HADOOP

Due to the challenges associated with configuring and running Hadoop on Windows, I installed Hadoop on **Ubuntu** within a **virtual machine (VM)**.

I downloaded Hadoop by following a step-by-step guide provided on **Medium**, a platform that helps data science students regardless of their course of study. Here is the link to the guide: [Installing Hadoop on Ubuntu: A Step-by-Step Guide]

By the end of the setup, I had achieved the following:

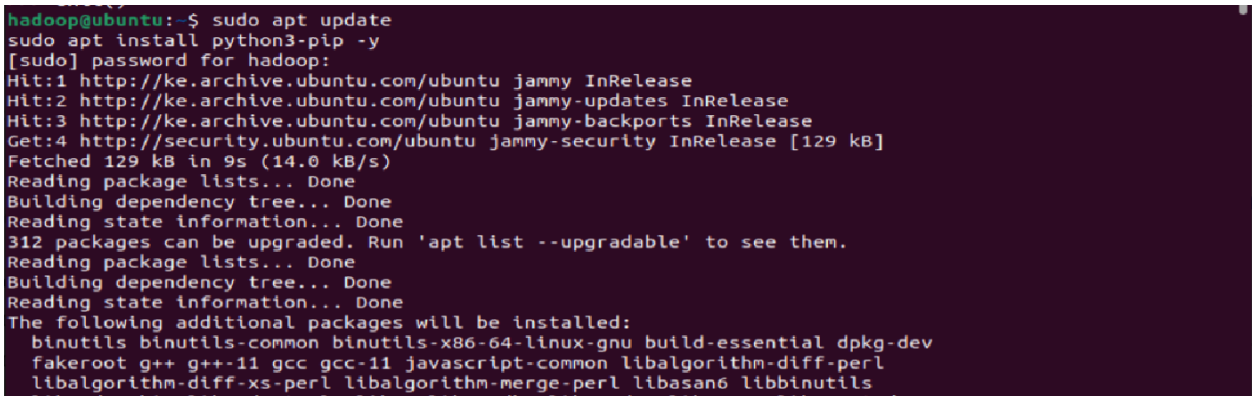
- I started Hadoop using the command , **" start-all.sh ".** This allowed me to start Hadoop after formatting the NameNode.
- I then used the **jps** command to check which Hadoop components were running

```
hadoop@ubuntu:~/hadoop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop 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' (ED25519) to the list of known hosts.
Starting resourcemanager
Starting nodemanagers
hadoop@ubuntu:~/hadoop$ jps
10770 ResourceManager
10277 NameNode
10376 DataNode
10877 NodeManager
10526 SecondaryNameNode
11006 Jps
```

- To verify if Hadoop was correctly installed, I used, **“ hadoop version “** . This returned the installed Hadoop version: **Hadoop 3.3.6**.
- Before installing Hadoop, I first installed and configured **Java**. To verify that Java was installed correctly, I used **“ java -version “** . This returned the installed Java version: **OpenJDK version 1.8.0_442**.

STEP 2: GET DATA FROM WEB API

- I first installed **Python** since it would be used in our data pipeline.



```
hadoop@ubuntu:~$ pip3 --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
hadoop@ubuntu:~$ pip3 install requests pandas
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (2.25.1)
Collecting pandas
  Downloading pandas-2.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.1 MB)
     ---------------------------------------- 13.1/13.1 MB 422.0 kB/s eta 0:00:00
Collecting tzdata>=2022.7
  Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
     ---------------------------------------- 346.8/346.8 KB 807.1 kB/s eta 0:00:00
Collecting python-dateutil>=2.8.2
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     ---------------------------------------- 229.9/229.9 KB 417.9 kB/s eta 0:00:00
Collecting numpy>=1.22.4
  Downloading numpy-2.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.4 MB)
     ---------------------------------------- 16.4/16.4 MB 1.9 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2
->pandas) (1.16.0)
Installing collected packages: tzdata, python-dateutil, numpy, pandas
  WARNING: The scripts f2py and numpy-config are installed in '/home/hadoop/.local/bin' which is not o
n PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-scr
ipt-location.
Successfully installed numpy-2.2.3 pandas-2.2.3 python-dateutil-2.9.0.post0 tzdata-2025.1
```

- I then loaded data from a web API, specifically from **randomuser.api**, and successfully retrieved **200 records**.

```
  GNU nano 6.2                                                  random_users.py
import requests
import mysql.connector

# Fetch Data from API
url = "https://randomuser.me/api/?results=10"  # Fetching 10 users for testing
response = requests.get(url)

if response.status_code == 200:
    data = response.json()["results"]

    user_data = []
    address_data = []
    login_data = []
    picture_data = []

    for user in data:
        # Users Table
        user_data.append((
            user["name"]["first"], user["name"]["last"],
            user["gender"], user["email"], user["phone"],
            user["location"]["city"], user["location"]["country"], user["dob"]["age"]
        ))

        # Address Table
        address_data.append((
            user["location"]["street"]["name"],
            user["location"]["city"], user["location"]["state"],
            user["location"]["country"], user["location"]["postcode"]
        ))

        # Login Table
        login_data.append((
            user["login"]["username"], user["login"]["password"],
            user["login"]["uuid"]
        ))
```

```
    # Save to CSV
    df.to_csv("random_users.csv", index=False)
    print("User data saved to 'random_users.csv' successfully!")
else:
    print(f"Failed to fetch data: {response.status_code}")
```

STEP 3 : INSTALL SQL

- After installing **MySQL**, I started the MySQL service using, **"sudo systemctl start mysql"**

```
hadoop@ubuntu:~$ sudo systemctl start mysql
sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
hadoop@ubuntu:~$ sudo systemctl status mysql
* mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:>
     Active: active (running) since Sat 2025-02-15 01:17:50 EAT; 47s ago
   Main PID: 14613 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 2149)
     Memory: 362.1M
        CPU: 1.598s
     CGroup: /system.slice/mysql.service
             `-14613 /usr/sbin/mysqld

Feb 15 01:17:48 ubuntu systemd[1]: Starting MySQL Community Server...
Feb 15 01:17:50 ubuntu systemd[1]: Started MySQL Community Server.
lines 1-13/13 (END)...skipping...
* mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2025-02-15 01:17:50 EAT; 47s ago
   Main PID: 14613 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 2149)
     Memory: 362.1M
        CPU: 1.598s
     CGroup: /system.slice/mysql.service
             `-14613 /usr/sbin/mysqld

Feb 15 01:17:48 ubuntu systemd[1]: Starting MySQL Community Server...
Feb 15 01:17:50 ubuntu systemd[1]: Started MySQL Community Server.
~
```

- I then loaded the retrieved API data into MySQL
- I create 4 tables from the retrieved API data. They include:

    1. Users data
    2. Login information
    3. Pictures
    4. Addresses

```
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE IF NOT EXISTS random_users_db;
Query OK, 1 row affected (0.29 sec)

mysql> USE random_users_db;
Database changed
mysql>
mysql> CREATE TABLE IF NOT EXISTS users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     gender VARCHAR(10),
    ->     email VARCHAR(100),
    ->     phone VARCHAR(20),
    ->     city VARCHAR(50),
    ->     country VARCHAR(50),
    ->     age INT
    -> );
Query OK, 0 rows affected (0.79 sec)

mysql> SET GLOBAL local_infile = 1;
Query OK, 0 rows affected (0.49 sec)

mysql> EXIT;
Bye
```

```
mysql> USE random_users_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE api_random_users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     gender VARCHAR(10),
    ->     email VARCHAR(100),
    ->     phone VARCHAR(20),
    ->     city VARCHAR(50),
    ->     country VARCHAR(50),
    ->     age INT
    -> );
Query OK, 0 rows affected (0.32 sec)

mysql> CREATE TABLE IF NOT EXISTS addresses (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     user_id INT,
    ->     street VARCHAR(100),
    ->     city VARCHAR(50),
    ->     state VARCHAR(50),
    ->     country VARCHAR(50),
    ->     postcode VARCHAR(20),
    ->     FOREIGN KEY (user_id) REFERENCES api_random_users(id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected, 1 warning (0.08 sec)

mysql> CREATE TABLE IF NOT EXISTS login_info (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     user_id INT,
    ->     username VARCHAR(50),
    ->     password VARCHAR(100),
    ->     uuid VARCHAR(50),
    ->     FOREIGN KEY (user_id) REFERENCES api_random_users(id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

- The following command is for loading the tables within the **“random_user_db”.**

```
mysql> SHOW TABLES;
+-----------------------------+
| Tables_in_random_users_db |
+-----------------------------+
| addresses                   |
| api_random_users            |
| login_info                  |
| pictures                    |
| users                       |
+-----------------------------+
5 rows in set (0.01 sec)

mysql> exit
Bye
```

- I changed the authentication plugin from **auth_socket** to **mysql_native_password** for user authentication.

```
hadoop@ubuntu:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 368
Server version: 8.0.41-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT user, host, plugin FROM mysql.user WHERE user = 'root';
+------+-----------+-------------+
| user | host      | plugin      |
+------+-----------+-------------+
| root | localhost | auth_socket |
+------+-----------+-------------+
1 row in set (0.07 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_new_password';
Query OK, 0 rows affected (0.30 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.10 sec)

mysql> EXIT;
Bye
```

- I granted the necessary privileges to the MySQL user.

```
hadoop@ubuntu:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 381
Server version: 8.0.41-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'hadoop123';
Query OK, 0 rows affected (0.09 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.04 sec)

mysql> EXIT;
Bye
```

- To confirm that the data was successfully stored, I ran a query for each table that selects 10 records:

    1. For user's data

The Users table contains the first name, last name, email address, phone number, city, country and age of different people.

```
mysql> SELECT * FROM users LIMIT 10;
+----+------------+-----------+--------+-------------------------------+---------------+-----------------+-------------+------+
| id | first_name | last_name | gender | email                         | phone         | city            | country     | age  |
+----+------------+-----------+--------+-------------------------------+---------------+-----------------+-------------+------+
|  1 | Leslie     | Bradley   | male   | leslie.bradley@example.com    | 071-253-8925  | Balbriggan      | Ireland     |   43 |
|  2 | Ava        | Edwards   | female | ava.edwards@example.com       | (824)-795-0912| Lower Hutt      | New Zealand |   49 |
|  3 | Aurélia    | Araújo    | female | aurelia.araujo@example.com    | (92) 6718-4362| Ribeirão Pires  | Brazil      |   31 |
|  4 | Diethelm   | Brehmer   | male   | diethelm.brehmer@example.com  | 0448-9144742  | Stade           | Germany     |   38 |
|  5 | Gustav     | Kristensen| male   | gustav.kristensen@example.com | 47221806      | Stenderup       | Denmark     |   49 |
|  6 | Nerea      | Vázquez   | female | nerea.vazquez@example.com     | 921-220-893   | Torrevieja      | Spain       |   65 |
|  7 | Alex       | Young     | female | alex.young@example.com        | 031-014-4157  | Trim            | Ireland     |   74 |
|  8 | Leon       | Brown     | male   | leon.brown@example.com        | 061-054-5270  | Killarney       | Ireland     |   76 |
|  9 | Hilla      | Salo      | female | hilla.salo@example.com        | 07-275-631    | Kalajoki        | Finland     |   50 |
| 10 | Paulo      | Mack      | male   | paulo.mack@example.com        | 0219-4306052  | Lahr/Schwarzwald| Germany     |   42 |
+----+------------+-----------+--------+-------------------------------+---------------+-----------------+-------------+------+
10 rows in set (0.01 sec)
```

    2. For addresses data

The address table contains the street, city, state, country and post code for each of the user in the first table of user table. Here we have **user_id** as our foreign key.

```
mysql>
mysql> -- Check Addresses Table
mysql> SELECT * FROM addresses LIMIT 10;
+----+---------+----------------------+-------------+----------------------+--------------+----------+
| id | user_id | street               | city        | state                | country      | postcode |
+----+---------+----------------------+-------------+----------------------+--------------+----------+
|  1 |      10 | Filistin Cd          | Amasya      | ?anl?urfa            | Turkey       | 47078    |
|  2 |       9 | Prospekt Peremozhciv | Zimogir'ya  | Zhitomirska          | Ukraine      | 68387    |
|  3 |       8 | Miodraga Nikoli?a    | ◆agubica    | Ma?va                | Serbia       | 92999    |
|  4 |       7 | Hebbrodweg           | Mill        | Friesland            | Netherlands  | 3657 CQ  |
|  5 |       6 | Doktorlar Cd         | Yalova      | ?anl?urfa            | Turkey       | 41318    |
|  6 |       5 | Rue de la Fontaine   | Petit-Val   | Glarus               | Switzerland  | 3579     |
|  7 |       4 | Continuación Cintrón | Tres Alamos | Campeche             | Mexico       | 54113    |
|  8 |       3 | Aleksanterinkatu     | Lumparland  | Northern Ostrobothnia| Finland      | 81231    |
|  9 |       2 | Rue de Gerland       | Vorderthal  | Zürich               | Switzerland  | 7226     |
| 10 |       1 | Pockrus Page Rd      | Provo       | Kentucky             | United States| 78289    |
+----+---------+----------------------+-------------+----------------------+--------------+----------+
10 rows in set (0.00 sec)
```

3. For Login information data

For the login information table, it contains also the username, password and uuid for each of the users in the user's table. Here we also have **user_id** as our foreign key.

```
mysql>
mysql> -- Check Login Info Table
mysql> SELECT * FROM login_info LIMIT 10;
+----+---------+----------------------+----------+--------------------------------------+
| id | user_id | username             | password | uuid                                 |
+----+---------+----------------------+----------+--------------------------------------+
|  1 |      10 | greensnake554        | 456654   | 95ab90c6-4328-44f2-a05d-83ba7a843c13 |
|  2 |       9 | heavypeacock267      | 13131313 | 54b5ee85-06f4-4b5b-ba53-ac32299d907c |
|  3 |       8 | orangelion211        | zurich   | 5eae47d9-1bc1-43fa-a43d-eaf93c8f4ee4 |
|  4 |       7 | tinybear873          | maxx     | 8e8ff881-ca84-4046-8a04-4f74c749b0e4 |
|  5 |       6 | goldenduck290        | scott    | ca65a707-95a8-4324-b7a5-9c69e0d38ed7 |
|  6 |       5 | tinyswan842          | savage   | a4ea4d21-5d90-42e8-92b5-cf2f6f899825 |
|  7 |       4 | greenfish281         | elijah   | 8c1aefe4-75f2-4fc5-b8d6-e0c2c5334959 |
|  8 |       3 | brownmeercat625      | superman | fa848c97-70e1-43ed-8227-c6d6aa50e876 |
|  9 |       2 | organicbutterfly628  | pleasure | c0cedd7e-a4cd-4733-94c6-1af7956bf506 |
| 10 |       1 | tinykoala221         | chrysler | 76dad397-d57e-4177-801d-b6356434a4e0 |
+----+---------+----------------------+----------+--------------------------------------+
10 rows in set (0.01 sec)
```

4. For Picture data

For the picture table it contains the images of the various users in the user table. Here we have also the **user_id** as our foreign key.

```
mysql> -- Check Pictures Table
mysql> SELECT * FROM pictures LIMIT 10;
+----+---------+-----------------------------------------------------+-----------------------------------------------------+----------
----------------------------------------------+
| id | user_id | large                                               | medium                                              | thumbnail
                                              |
+----+---------+-----------------------------------------------------+-----------------------------------------------------+----------
----------------------------------------------+
|  1 |      10 | https://randomuser.me/api/portraits/men/64.jpg      | https://randomuser.me/api/portraits/med/men/64.jpg  | https://r
andomuser.me/api/portraits/thumb/men/64.jpg   |
|  2 |       9 | https://randomuser.me/api/portraits/men/91.jpg      | https://randomuser.me/api/portraits/med/men/91.jpg  | https://r
andomuser.me/api/portraits/thumb/men/91.jpg   |
|  3 |       8 | https://randomuser.me/api/portraits/women/52.jpg    | https://randomuser.me/api/portraits/med/women/52.jpg | https://r
andomuser.me/api/portraits/thumb/women/52.jpg |
|  4 |       7 | https://randomuser.me/api/portraits/men/90.jpg      | https://randomuser.me/api/portraits/med/men/90.jpg  | https://r
andomuser.me/api/portraits/thumb/men/90.jpg   |
|  5 |       6 | https://randomuser.me/api/portraits/women/30.jpg    | https://randomuser.me/api/portraits/med/women/30.jpg | https://r
andomuser.me/api/portraits/thumb/women/30.jpg |
|  6 |       5 | https://randomuser.me/api/portraits/men/67.jpg      | https://randomuser.me/api/portraits/med/men/67.jpg  | https://r
andomuser.me/api/portraits/thumb/men/67.jpg   |
|  7 |       4 | https://randomuser.me/api/portraits/women/27.jpg    | https://randomuser.me/api/portraits/med/women/27.jpg | https://r
andomuser.me/api/portraits/thumb/women/27.jpg |
|  8 |       3 | https://randomuser.me/api/portraits/men/1.jpg       | https://randomuser.me/api/portraits/med/men/1.jpg   | https://r
andomuser.me/api/portraits/thumb/men/1.jpg     |
|  9 |       2 | https://randomuser.me/api/portraits/women/85.jpg    | https://randomuser.me/api/portraits/med/women/85.jpg | https://r
andomuser.me/api/portraits/thumb/women/85.jpg |
| 10 |       1 | https://randomuser.me/api/portraits/women/49.jpg    | https://randomuser.me/api/portraits/med/women/49.jpg | https://r
andomuser.me/api/portraits/thumb/women/49.jpg |
+----+---------+-----------------------------------------------------+-----------------------------------------------------+----------
----------------------------------------------+
10 rows in set (0.00 sec)
```

STEP 4: DOWNLOAD AND CONFIGURE SQOOP

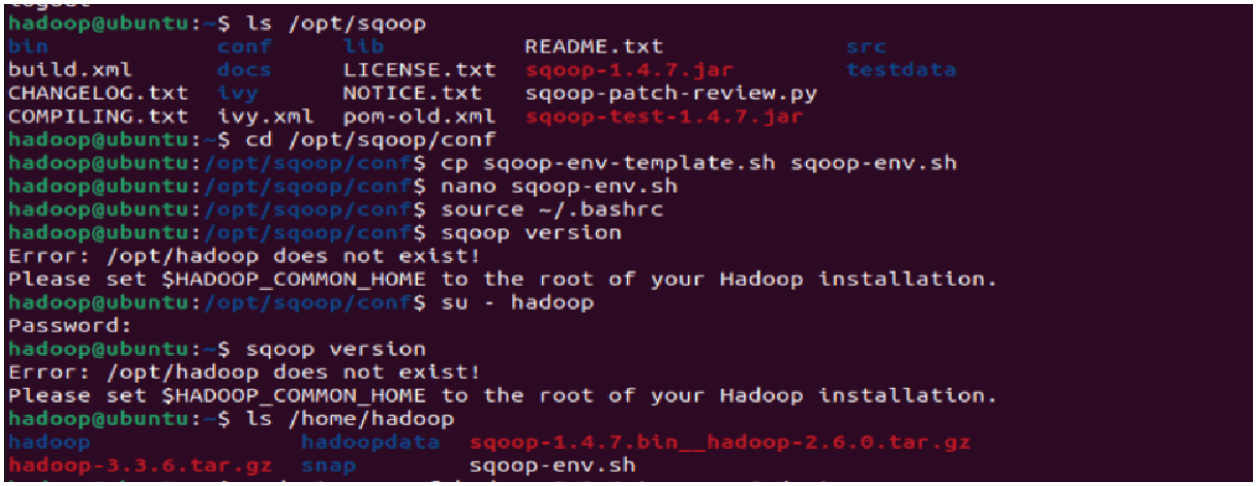
- I installed **Sqoop** by downloading it from the official Apache website and extracted the files.

```
ubuntu@ubuntu:~$ su - hadoop
Password:
hadoop@ubuntu:~$ wget https://downloads.apache.org/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
--2025-02-15 00:43:45--  https://downloads.apache.org/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f8:10a:39da::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://attic.apache.org/projects/sqoop.html [following]
--2025-02-15 00:43:46--  http://attic.apache.org/projects/sqoop.html
Resolving attic.apache.org (attic.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to attic.apache.org (attic.apache.org)|151.101.2.132|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://attic.apache.org/projects/sqoop.html [following]
--2025-02-15 00:43:47--  https://attic.apache.org/projects/sqoop.html
Connecting to attic.apache.org (attic.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12574 (12K) [text/html]
Saving to: 'sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz'

sqoop-1.4.7.bin__ha 100%[===================>]  12.28K  --.-KB/s    in 0.02s

2025-02-15 00:43:47 (677 KB/s) - 'sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz' saved [12574/12574]

hadoop@ubuntu:~$ tar -xzvf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz

gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error is not recoverable: exiting now
hadoop@ubuntu:~$ wget https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
--2025-02-15 00:44:31--  https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17953604 (17M) [application/x-gzip]
Saving to: 'sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz.1'

sqoop-1.4.7.bin__ha 100%[===================>]  17.12M  1.76MB/s    in 9.9s
```

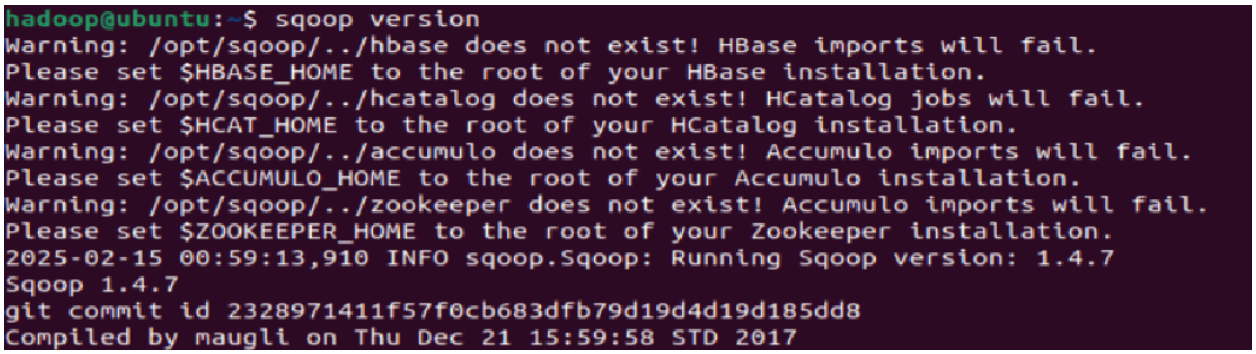
- To verify the installation, I used, **“ ls -l “**. This confirmed that **Sqoop** was successfully installed.

```
hadoop@ubuntu:~$ ls -l
total 730556
drwxr-xr-x 11 hadoop hadoop      4096 Feb 15 00:33 hadoop
-rw-rw-r--  1 hadoop hadoop 730107476 Jun 26  2023 hadoop-3.3.6.tar.gz
drwxrwxr-x  3 hadoop hadoop      4096 Feb 15 00:31 hadoopdata
drwx------  3 hadoop hadoop      4096 Feb 15 00:18 snap
drwxr-xr-x  9 hadoop hadoop      4096 Des 19  2017 sqoop-1.4.7.bin__hadoop-2.6.0
-rw-rw-r--  1 hadoop hadoop  17953604 Jul  6  2020 sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
-rw-rw-r--  1 hadoop hadoop         1 Feb 15 00:47 sqoop-env.sh
```

- I then checked if Sqoop was properly integrated within my Hadoop home directory using **“ ls /home/hadoop”**



- Finally, I verified the installed version of Sqoop using, **“ sqoop version “.** This returned: **Sqoop 1.4.7.**



STEP 5: INSTALL JDBC DRIVER

- I installed the **JDBC (Java Database Connector)** since it is required to connect MySQL to Sqoop.
- To verify that JDBC was successfully installed, I used, **“ ls -l “.** This confirmed the presence of the file **mysql-connector-java-8.0.23.tar.gz.**
- I then navigated to the directory where the file was extracted and confirmed the existence of **mysql-connector-java-8.0.23.jar**.

```
hadoop@ubuntu:~$ sudo dpkg -i /home/ubuntu/Downloads/mysql-connector-j_9.2.0-1ubuntu22.04_all.deb
Selecting previously unselected package mysql-connector-j.
(Reading database ... 205543 files and directories currently installed.)
Preparing to unpack .../mysql-connector-j_9.2.0-1ubuntu22.04_all.deb ...
Unpacking mysql-connector-j (9.2.0-1ubuntu22.04) ...
Setting up mysql-connector-j (9.2.0-1ubuntu22.04) ...
hadoop@ubuntu:~$ dpkg -L mysql-connector-j | grep '\.jar$'
/usr/share/java/mysql-connector-j-9.2.0.jar
/usr/share/java/mysql-connector-java-9.2.0.jar
hadoop@ubuntu:~$ sudo cp /usr/share/java/mysql-connector-java-9.2.0.jar /usr/lib/sqoop/lib/
cp: cannot create regular file '/usr/lib/sqoop/lib/': No such file or directory
hadoop@ubuntu:~$ sudo cp /usr/share/java/mysql-connector-java-9.2.0.jar /opt/sqoop/lib/
hadoop@ubuntu:~$ ls -lh /opt/sqoop/lib/mysql-connector-java-9.2.0.jar
-rw-r--r-- 1 root root 2.5M Feb 15 10:41 /opt/sqoop/lib/mysql-connector-java-9.2.0.jar
```

```
hadoop@ubuntu:~$ ls -l
total 738444
drwxr-xr-x 11 hadoop hadoop      4096 Feb 15 00:33 hadoop
-rw-rw-r--  1 hadoop hadoop 730107476 Jun 26  2023 hadoop-3.3.6.tar.gz
drwxrwxr-x  3 hadoop hadoop      4096 Feb 15 00:31 hadoopdata
drwxr-xr-x  3 hadoop hadoop      4096 Des  1  2020 mysql-connector-java-8.0.23
-rw-rw-r--  1 hadoop hadoop   4036068 Des  1  2020 mysql-connector-java-8.0.23.tar.gz
-rw-rw-r--  1 hadoop hadoop   4036068 Des  1  2020 mysql-connector-java-8.0.23.tar.gz.1
drwx------  3 hadoop hadoop      4096 Feb 15 00:18 snap
-rw-rw-r--  1 hadoop hadoop  17953604 Jul  6  2020 sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
-rw-rw-r--  1 hadoop hadoop         1 Feb 15 00:47 sqoop-env.sh
hadoop@ubuntu:~$ cd mysql-connector-java-8.0.23
hadoop@ubuntu:~/mysql-connector-java-8.0.23$ ls -l
total 2828
-rw-r--r-- 1 hadoop hadoop   88878 Des  1  2020 build.xml
-rw-r--r-- 1 hadoop hadoop  267994 Des  1  2020 CHANGES
-rw-r--r-- 1 hadoop hadoop     186 Des  1  2020 INFO_BIN
-rw-r--r-- 1 hadoop hadoop     136 Des  1  2020 INFO_SRC
-rw-r--r-- 1 hadoop hadoop  100771 Des  1  2020 LICENSE
-rw-r--r-- 1 hadoop hadoop 2415211 Des  1  2020 mysql-connector-java-8.0.23.jar
-rw-r--r-- 1 hadoop hadoop    1245 Des  1  2020 README
drwxr-xr-x 8 hadoop hadoop    4096 Des  1  2020 src
```

- To ensure that the JDBC driver was properly added to the **Sqoop libraries**, I ran , **“ ls /opt/sqoop/lib “**

```
hadoop@ubuntu:~/mysql-connector-java-8.0.23$ ls /opt/sqoop/lib/
ant-contrib-1.0b3.jar            kite-data-hive-1.1.0.jar
ant-eclipse-1.0-jvm1.2.jar       kite-data-mapreduce-1.1.0.jar
avro-1.8.1.jar                   kite-hadoop-compatibility-1.1.0.jar
avro-mapred-1.8.1-hadoop2.jar    mysql-connector-java-8.0.23.jar
commons-codec-1.4.jar            opencsv-2.3.jar
commons-compress-1.8.1.jar       paranamer-2.7.jar
commons-io-1.4.jar               parquet-avro-1.6.0.jar
commons-jexl-2.1.1.jar           parquet-column-1.6.0.jar
commons-lang3-3.4.jar            parquet-common-1.6.0.jar
commons-logging-1.1.1.jar        parquet-encoding-1.6.0.jar
hsqldb-1.8.0.10.jar              parquet-format-2.2.0-rc1.jar
jackson-annotations-2.3.1.jar    parquet-generator-1.6.0.jar
jackson-core-2.3.1.jar           parquet-hadoop-1.6.0.jar
jackson-core-asl-1.9.13.jar      parquet-jackson-1.6.0.jar
jackson-databind-2.3.1.jar       slf4j-api-1.6.1.jar
jackson-mapper-asl-1.9.13.jar    snappy-java-1.1.1.6.jar
kite-data-core-1.1.0.jar         xz-1.5.jar
```

STEP 6: TRANSFER THE WEB API DATA TO HDFS USING SQOOP

- The data stored in MySQL was transferred to **HDFS** using **Sqoop**

```
hadoop@ubuntu:~$ sqoop import \
--connect "jdbc:mysql://localhost/random_users_db" \
--username root \
-P \
--table users \
--target-dir /user/hdfs/random_users \
--fields-terminated-by ',' \
--as-textfile \
--num-mappers 1 \
--delete-target-dir \
--direct
Warning: /opt/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /opt/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /opt/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /opt/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2025-02-17 20:02:10,236 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
Enter password:
2025-02-17 20:02:13,715 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
```

- After the data was in HDFS, I performed a **MapReduce** job for processing.

```
2025-02-17 20:02:50,257 INFO mapred.LocalJobRunner: Finishing task: attempt_local933432142_0001_m_0000
00_0
2025-02-17 20:02:50,262 INFO mapred.LocalJobRunner: map task executor complete.
2025-02-17 20:02:50,365 INFO mapreduce.Job:  map 100% reduce 0%
2025-02-17 20:02:50,366 INFO mapreduce.Job: Job job_local933432142_0001 completed successfully
2025-02-17 20:02:50,512 INFO mapreduce.Job: Counters: 21
        File System Counters
                FILE: Number of bytes read=10681
                FILE: Number of bytes written=664316
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=0
                HDFS: Number of bytes written=826
                HDFS: Number of read operations=7
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=1
                Map output records=10
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=62
                Total committed heap usage (bytes)=83443712
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=826
2025-02-17 20:02:50,527 INFO mapreduce.ImportJobBase: Transferred 826 bytes in 17.8677 seconds (46.228
5 bytes/sec)
2025-02-17 20:02:50,532 INFO mapreduce.ImportJobBase: Retrieved 10 records.
hadoop@ubuntu:~$
```

- To verify that the data was successfully loaded into HDFS, I ran a command to print the reduced task output.

```
hadoop@ubuntu:~$ hadoop fs -ls /user/hdfs/random_users
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-02-17 20:02 /user/hdfs/random_users/_SUCCESS
-rw-r--r--   1 hadoop supergroup        826 2025-02-17 20:02 /user/hdfs/random_users/part-m-00000
hadoop@ubuntu:~$ hdfs dfs -cat /user/hdfs/random_users/part-m-00000
1,Leslie,Bradley,male,leslie.bradley@example.com,071-253-8925,Balbriggan,Ireland,43
2,Ava,Edwards,female,ava.edwards@example.com,(824)-795-0912,Lower Hutt,New Zealand,49
3,AurÃ©lia,AraÃºjo,female,aurelia.araujo@example.com,(92) 6718-4362,RibeirÃ£o Pires,Brazil,31
4,Diethelm,Brehmer,male,diethelm.brehmer@example.com,0448-9144742,Stade,Germany,38
5,Gustav,Kristensen,male,gustav.kristensen@example.com,47221806,Stenderup,Denmark,49
6,Nerea,VÃ¡zquez,female,nerea.vazquez@example.com,921-220-893,Torrevieja,Spain,65
7,Alex,Young,female,alex.young@example.com,031-014-4157,Trim,Ireland,74
8,Leon,Brown,male,leon.brown@example.com,061-054-5270,Killarney,Ireland,76
9,Hilla,Salo,female,hilla.salo@example.com,07-275-631,Kalajoki,Finland,50
10,Paulo,Mack,male,paulo.mack@example.com,0219-4306052,Lahr/Schwarzwald,Germany,42
hadoop@ubuntu:~$
```

STEP 7 : TRANSFER DATA FROM HDFS TO SQL

- Finally, the processed data from **HDFS** was transferred back to **MySQL** for storage.

```
hadoop@ubuntu:~$ sudo /bin/chmod 777 /home/ubuntu
hadoop@ubuntu:~$ hadoop fs -get /user/hdfs/random_users/part-m-00000 /home/ubuntu/random_users.txt
hadoop@ubuntu:~$ python3
Python 3.10.12 (main, Jan 17 2025, 14:35:34) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>>
>>> # Read the text file (as it appears to be comma-separated)
>>> df = pd.read_csv('/home/ubuntu/random_users.txt', header=None)
>>>
>>> # Add column names
>>> df.columns = ['id', 'first_name', 'last_name', 'gender', 'email', 'phone', 'city', 'country', 'age
']
>>>
>>> # Write the DataFrame to a CSV file
>>> df.to_csv('/home/ubuntu/random_users.csv', index=False)
>>>
>>> print("CSV file created successfully!")
CSV file created successfully!
>>> exit()
```

MapReduce

Create a MapReduce -based implementation for the following tasks:

1. Line Length Counter

- Input: A text file.

```
nano line_length_mapper.py
```

```
#!/usr/bin/env python3
import sys

def main():
    for i, line in enumerate(sys.stdin, start=1):
        print(f"{i}\t{len(line.strip())}")

if __name__ == "__main__":
```

```
    main()
```

```
nano line_length_reducer.py
```

```
#!/usr/bin/env python3
import sys

def main():
    for line in sys.stdin:
        print(line.strip())

if __name__ == "__main__":
    main()
```

```
hadoop@ubuntu:~$ hdfs dfs -mkdir -p /user/ubuntu/input
hadoop@ubuntu:~$ hdfs dfs -put input.txt /user/ubuntu/input
```

- Mapper: Emits (line_number, length_of_line).
- Reducer: Passes through the values as they are

```
hadoop@ubuntu:~$ find / -name "hadoop-streaming*.jar" 2>/dev/null
/home/hadoop/hadoop/share/hadoop/tools/sources/hadoop-streaming-3.3.6-sources.ja
r
/home/hadoop/hadoop/share/hadoop/tools/sources/hadoop-streaming-3.3.6-test-sourc
es.jar
/home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar
hadoop@ubuntu:~$ hadoop jar /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-st
reaming-3.3.6.jar \
    -input /user/ubuntu/input \
    -output /user/ubuntu/output1 \
    -mapper "python3 line_length_mapper.py" \
    -reducer "python3 line_length_reducer.py"
2025-02-19 08:20:32,635 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-02-19 08:20:33,220 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
```

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=98
                HDFS: Number of bytes written=15
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Map output bytes=15
                Map output materialized bytes=27
                Input split bytes=101
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=27
                Reduce input records=3
                Reduce output records=3
                Spilled Records=6
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=138
                Total committed heap usage (bytes)=270671872
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=49
        File Output Format Counters
                Bytes Written=15
2025-02-19 08:20:41,878 INFO streaming.StreamJob: Output directory: /user/ubuntu
/output1
hadoop@ubuntu:~$
```

- Output: Line number with its length.

```
hadoop@ubuntu:~$ hdfs dfs -cat /user/ubuntu/output1/part-00000
1       11
2       15
3       20
```

2. Word Co-Occurrence Matrix

- Input: A text file.

```
nano word_cooccurrence_mapper.py
```

```
#!/usr/bin/env python3
import sys
```

```
from itertools import combinations

def main():
for line in sys.stdin:
words = line.strip().split()
for word1, word2 in combinations(words, 2):
print(f"{word1},{word2}\t1")

if __name__ == "__main__":
main()
```

```
nano word_cooccurrence_reducer.py
```

```
#!/usr/bin/env python3
import sys
from collections import defaultdict

def main():
word_pairs = defaultdict(int)
for line in sys.stdin:
pair, count = line.strip().split('\t')
word_pairs[pair] += int(count)

for pair, count in word_pairs.items():
print(f"{pair}\t{count}")

if __name__ == "__main__":
main()
```

```
hadoop@ubuntu:~$ nano sentiment_mapper.py
hadoop@ubuntu:~$ nano sentiment_reducer.py
hadoop@ubuntu:~$ chmod +x sentiment_mapper.py sentiment_reducer.py
hadoop@ubuntu:~$ hadoop jar    /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-
```

- Mapper: Emits pairs of words that appear together.
- Reducer: Counts the number of times each pair occurs.

```
hadoop@ubuntu:~$ hadoop jar   /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-
streaming-3.3.6.jar \
    -input /user/ubuntu/sentiment_input \
    -output /user/ubuntu/sentiment_output \
    -mapper "python3 sentiment_mapper.py" \
    -reducer "python3 sentiment_reducer.py"
2025-02-19 08:42:49,059 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-02-19 08:42:49,594 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
```

```
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=98
                HDFS: Number of bytes written=127
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=3
                Map output records=10
                Map output bytes=127
                Map output materialized bytes=153
                Input split bytes=101
                Combine input records=0
                Combine output records=0
                Reduce input groups=10
                Reduce shuffle bytes=153
                Reduce input records=10
                Reduce output records=10
                Spilled Records=20
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=150
                Total committed heap usage (bytes)=270671872
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=49
        File Output Format Counters
                Bytes Written=127
2025-02-19 08:30:48,450 INFO streaming.StreamJob: Output directory: /user/ubuntu
/output2
```

- Output: A matrix showing word co-occurrences.

```
hadoop@ubuntu:~$ hdfs dfs -cat /user/ubuntu/output2/part-00000
Big,data        1
Big,is  1
Big,powerful    1
Hadoop,great    1
Hadoop,is       1
Hello,world     1
data,is 1
data,powerful   1
is,great        1
is,powerful     1
hadoop@ubuntu:~$
```

3. Sentiment Analysis on Tweets

- Input: A dataset of tweets.

```
nano sentiment_mapper.py
```

```
#!/usr/bin/env python3
import sys

sentiment_dict = {
    "happy": "positive",
    "good": "positive",
    "great": "positive",
    "bad": "negative",
    "sad": "negative",
    "angry": "negative",
    "ok": "neutral",
    "fine": "neutral"
}

def main():
    for line in sys.stdin:
        words = line.strip().split()
        for word in words:
            sentiment = sentiment_dict.get(word.lower(), "neutral")
            print(f"{sentiment}\t1")

if __name__ == "__main__":
    main()
```

```
nano sentiment_reducer.py
```

```python
#!/usr/bin/env python3
import sys
from collections import defaultdict

def main():
    sentiment_counts = defaultdict(int)
    for line in sys.stdin:
        sentiment, count = line.strip().split('\t')
        sentiment_counts[sentiment] += int(count)

    for sentiment, count in sentiment_counts.items():
        print(f"{sentiment}\t{count}")

if __name__ == "__main__":
    main()
```

```
hadoop@ubuntu:~$ nano sentiment_mapper.py
hadoop@ubuntu:~$ nano sentiment_reducer.py
hadoop@ubuntu:~$ chmod +x sentiment_mapper.py sentiment_reducer.py
hadoop@ubuntu:~$ hadoop jar    /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-
```

- Mapper: Assigns sentiment scores (e.g., positive, negative, neutral) using a dictionary.
- Reducer: Aggregates sentiment scores per category.

```
hadoop@ubuntu:~$ hadoop jar    /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-
streaming-3.3.6.jar \
    -input /user/ubuntu/sentiment_input \
    -output /user/ubuntu/sentiment_output \
    -mapper "python3 sentiment_mapper.py" \
    -reducer "python3 sentiment_reducer.py"
2025-02-19 08:42:49,059 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-02-19 08:42:49,594 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
```

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=402
                HDFS: Number of bytes written=33
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=6
                Map output records=43
                Map output bytes=436
                Map output materialized bytes=528
                Input split bytes=121
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=528
                Reduce input records=43
                Reduce output records=3
                Spilled Records=86
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=199
                Total committed heap usage (bytes)=270671872
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=201
        File Output Format Counters
                Bytes Written=33
2025-02-19 08:42:58,860 INFO streaming.StreamJob: Output directory: /user/ubuntu
/sentiment_output
```

- Output: A sentiment distribution report.

```
hadoop@ubuntu:~$ hdfs dfs -cat /user/ubuntu/sentiment_output/part-00000
negative        3
neutral 37
positive        3
hadoop@ubuntu:~$
```