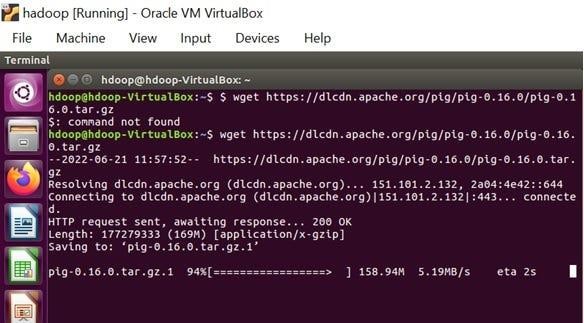
# EXP 4: Create UDF in PIG

**Step-by-step installation of Apache Pig on Hadoop cluster on Ubuntu Pre-requisite:**

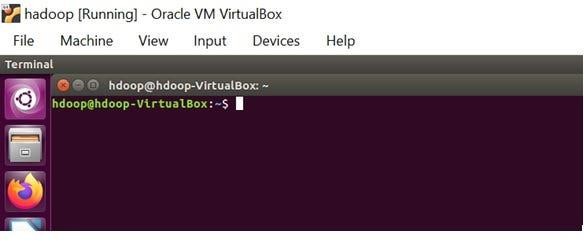
* Ubuntu 16.04 or higher version running (I have installed Ubuntu on Oracle VM (Virtual Machine) VirtualBox),
* Run Hadoop on ubuntu (I have installed Hadoop 3.2.1 on Ubuntu 16.04). You may refer to my blog “How to install Hadoop installation” click [here](https://medium.com/mlearning-ai/installation-of-apache-hadoop-3-2-1-on-ubuntu-67073ce208d7) for Hadoop installation).

**Pig installation steps Step 1:** Login into Ubuntu



**Step 2**: Go to <https://pig.apache.org/releases.html>and copy the path of the latest version of pig that you want to install. Run the following comment to download Apache Pig in Ubuntu:

$ wget <https://dlcdn.apache.org/pig/pig-0.16.0/pig-0.16.0.tar.gz>



**Step 3**: To untar pig-0.16.0.tar.gz file run the following command:

$ tar xvzf pig-0.16.0.tar.gz

**Step 4:** To create a pig folder and move pig-0.16.0 to the pig folder, execute the following command:

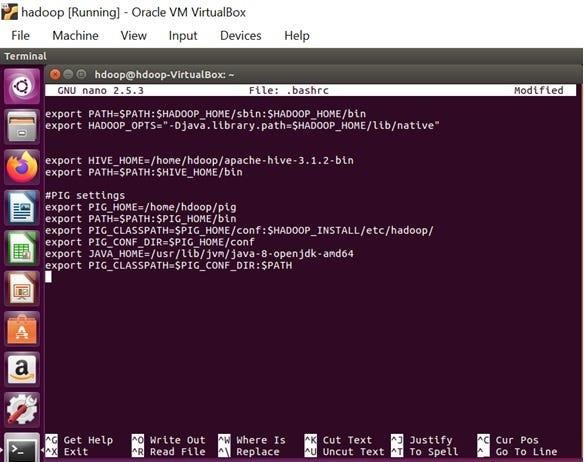
$ sudo mv /home/hdoop/pig-0.16.0 /home/hdoop/pig

**Step 5:** Now open the .bashrc file to edit the path and variables/settings for pig. Run the following command:

$ sudo nano .bashrc

Add the below given to .bashrc file at the end and save the file.

#PIG settingsexport PIG\_HOME=/home/hdoop/pigexport PATH=$PATH:$PIG\_HOME/binexport PIG\_CLASSPATH=$PIG\_HOME/conf:$HADOOP\_INSTALL/etc/hadoop/export PIG\_CONF\_DIR=$PIG\_HOME/confexport JAVA\_HOME=/usr/lib/jvm/java-8-openjdk- amd64export PIG\_CLASSPATH=$PIG\_CONF\_DIR:$PATH#PIG setting ends



**Step 6:** Run the following command to make the changes effective in the .bashrc file:

$ source .bashrc

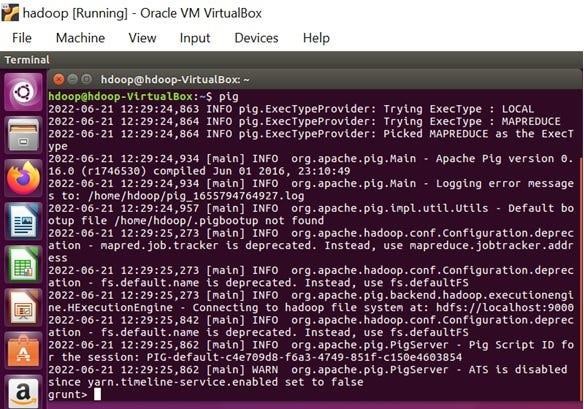
**Step 7:** To start all Hadoop daemons, navigate to the hadoop-3.2.1/sbin folder and run the following commands:

$ ./start-dfs.sh$ ./start-yarn$ jps



**Step 8:** Now you can launch pig by executing the following command:

$ pig



**Step 9:** Now you are in pig and can perform your desired tasks on pig. You can come out of the pig by the quit command:

> quit;

# CREATE USER DEFINED FUNCTION(UDF)

**Aim** : To create User Define Function in Apache Pig and execute it on map reduce.

# Procedure:

**Create a sample text file** hadoop@Ubuntu:~/Documents$ nano sample.txt Paste the below content to sample.txt

1,John 2,Jane 3,Joe 4,Emma

hadoop@Ubuntu:~/Documents$ hadoop fs -put sample.txt /home/hadoop/piginput/

# Create PIG File

hadoop@Ubuntu:~/Documents$ nano demo\_pig.pig

# paste the below the content to demo\_pig.pig

-- Load the data from HDFS

data = LOAD '/home/hadoop/piginput/sample.txt' USING PigStorage(',') AS (id:int>

-- Dump the data to check if it was loaded correctly DUMP data;

# Run the above file

hadoop@Ubuntu:~/Documents$ pig demo\_pig.pig

2024-08-07 12:13:08,791 [main] INFO

org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil

- Total input paths to process : 1 (1,John)

(2,Jane)

(3,Joe)

(4,Emma)

**Create udf file an save as uppercase\_udf.py**

uppercase\_udf.py

def uppercase(text): return text.upper()

if name == " main ": import sys

for line in sys.stdin:

line = line.strip()

result = uppercase(line) print(result)

# Create the udfs folder on hadoop

**hadoop@Ubuntu:~/Documents$ hadoop fs -mkdir /home/hadoop/udfs**

# put the upppercase\_udf.py in to the abv folder

**hadoop@Ubuntu:~/Documents$ hdfs dfs -put uppercase\_udf.py /home/hadoop/udfs/**

# hadoop@Ubuntu:~/Documents$ nano udf\_example.pig copy and paste the below content on udf\_example.pig

-- Register the Python UDF script

REGISTER 'hdfs:///home/hadoop/udfs/uppercase\_udf.py' USING jython AS udf;

-- Load some data

data = LOAD 'hdfs:///home/hadoop/sample.txt' AS (text:chararray);

-- Use the Python UDF

uppercased\_data = FOREACH data GENERATE udf.uppercase(text) AS uppercase\_text;

-- Store the result

STORE uppercased\_data INTO 'hdfs:///home/hadoop/pig\_output\_data';

# place sample.txt file on hadoop

hadoop@Ubuntu:~/Documents$ hadoop fs -put sample.txt /home/hadoop/

# To Run the pig file

hadoop@Ubuntu:~/Documents$ pig -f udf\_example.pig

# finally u get Success!

**Job Stats (time in seconds):**

JobId Maps Reduces MaxMapTimeMinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime

Alias Feature Outputs

job\_local1786848041\_0001 1 0 n/a n/a n/a n/a 00 0 0 data,uppercased\_data MAP\_ONLY hdfs:///home/hadoop/pig\_output\_data,

Input(s):

Successfully read 4 records (42778068 bytes) from: "hdfs:///home/hadoop/sample.txt"

Output(s):

Successfully stored 4 records (42777870 bytes) in: "hdfs:///home/hadoop/pig\_output\_data"

Counters:

Total records written : 4

Total bytes written : 42777870

Spillable Memory Manager spill count : 0 Total bags proactively spilled: 0

Total records proactively spilled: 0

Job DAG:

job\_local1786848041\_0001

2024-08-07 13:33:04,631 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!

2024-08-07 13:33:04,639 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!

2024-08-07 13:33:04,644 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!

2024-08-07 13:33:04,667 [main] INFO

org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!

# Note :

**If any error check jython package is installed and check the path specified on the above steps are give correctly**

# --------------------------------------------------------------------------------------------------------------

**To check the output file is created**

hadoop@Ubuntu:~/Documents$ hdfs dfs -ls /home/hadoop/pig\_output\_data Found 2 items

If you need to examine the files in the output folder, use:

# To view the output

**hadoop@Ubuntu:~/Documents$ hdfs dfs -cat /home/hadoop/pig\_output\_data/part-m- 00000**

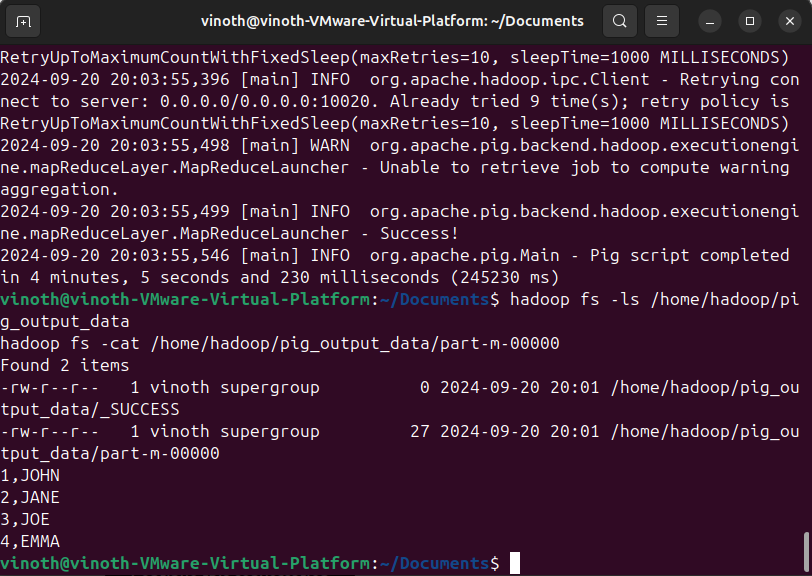
1,JOHN

2,JANE

3,JOE

4,EMMA

**Output:**

****

# Result:

Thus the program is executed successfully