SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY, TUMAKURU.

(A Constituent College of Sri Siddhartha Academy of Higher Education, Agalakote, Tumakuru)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY MANUAL

for

MCA

III SEMESTER

22MCA3LB2: BIG DATA LAB

Prepared by:

Dr.Kavyashree N
Assistant Professor
Dept of MCA, SSIT

PREFACE

A good basic knowledge is a strong foundation for every aspect. Advanced Database and Bigdata Laboratory provides an environment for learning and better understanding of the basic concepts and implementation of PL/SQL programming and Hadoop Ecosystem. The students will be able to enhance their knowledge in large and distributed information system.

Dr. M Siddappa,
Professor & Head,
Dept. of CSE, SSIT

VISION OF THE INSTITUTE

To carve technically competent, confident and socially responsible engineers.

MISSION OF THE INSTITUTE

- To impart fundamental knowledge in science and technology.
- To create a conducive ambience for better learning and to bring out creativity in the students.
- To instil managerial, entrepreneurial and soft skills.
- To evolve as trusted destination for quality technical education.
- Positive contribution to meet societal needs.
- To inculcate a spirit of enquiry, make learning perceptive and rational.

VISION OF THE DEPARTMENT

To craft professionally skilled engineers with research orientation, innovative insights and a passion for life-long learning to meet the needs of Industry and Society.

MISSION OF THE DEPARTMENT

M1: To offer need-based curriculum in collaboration with industry.

M2: To inculcate professional skills with innovative thinking to address societal problems of multidisciplinary nature.

M3: To provide a congenial environment to learn and exhibit soft skills.

M4: To promote research culture and the need for life-long learning.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO1: Excel in professional career and higher education by acquiring knowledge in mathematical, computing and engineering principles.

PEO2: Analyse societal problems and provide technically competent solutions.

PEO3: Possess academic excellence through innovative insights, soft skills and life-long learning.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Demonstrate the uses of knowledge by writing programs and integrate them with hardware/software products in multidisciplinary environment.

PSO2: Participate in planning and implementation of solutions to cater the industry specific requirements.

PROGRAM OUTCOMES

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems

PO2: Problem analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess Societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

GUIDELINES FOR CONTINUOUS INTERNAL EVALUATION AND SUGGESTED LAB RUBRICS

Laboratory Course Evaluation:

The distribution of marks for laboratory courses evaluation is shown in the following table:

Assessment tools			Marks	Total marks	Weightage for CO attainment
CIE	Lab Test1		10	50	50%
	Lab Test2		10		
	Weekly internal evaluation		30		
SEE	Lab End-term examination	Procedure write up	10	50	50%
		Execution of Part A and Part B	30		
		Viva voce	10		
	Total Marks			100	100%

Table 1. Laboratory Course Evaluation

Students' performance in Laboratory course is evaluated using rubrics defined by the program.

Rubrics for evaluating a lab test in programming lab are displayed in the following table.

Sl. No	Evaluation Parameters	Excellent	Very Good	Good	Satisfactory
1	Writing Program **[Max. Marks 5]**	5 Marks	4 Marks	3 Marks	2 Marks
		Completeness of code, consistent variable naming and formatting, well commented, uses existing skills in new	Completeness of code, consistent variable naming and formatting, lack of comments, uses existing skills in	Completeness of code, inconsistent variable naming and formatting, lacks clarity in commenting,	Lack of completeness of code, improper variable naming and not formatted, lacks comments, uses

		ways/learns new skills to solve the experimental problem.	new ways/learns new skills to solve the experimental problem.	uses existing skills to solve the experimental problem.	existing skills to partially solve the experimental problem.
2	Program Execution **[Max. Marks 5]**	5 Marks	4 Marks	3 Marks	1Marks
		Program is free of errors and output is well formatted. Demonstrates excellent problem solving and creativity skills.	Program is free of errors and output is well formatted. Demonstrates better problem solving and creativity skills.	Program is free of errors and output is not properly formatted. Demonstrates a clear understanding of the concepts relevant to the experiment.	Program contains few logical errors and output is not formatted. Demonstrates partial understanding of the concepts relevant to the experiment.
3	Program Modification and verify the results **[Max. Marks 5]**	5Marks	4Marks	3Marks	1 Marks
		Able to modify the changes specified and test the output.	Able to modify the changes specified with help could test the output.	Able to modify the changes specified with help could test the output.	Unable to modify the changes specified and unable to test the output.
4	Viva-Voice **[Max. Marks 5]**	**5Marks**	**4Marks**	**3Marks**	**1 Marks**
		Answers all the viva questions.	Answer 80% of the viva questions.	Answer 50% of the viva questions.	Couldn't answer viva questions properly.

Table 2. Rubrics for Lab Test Evaluation

Students' ability in developing programs, and testing the program, documenting the work done through lab record, etc are evaluated continuously during all lab sessions.

Sl. No	**Evaluation Component**	**Excellent**	**Good**	**Fair**	**Satisfactory**
1	Writing Program [Max. Marks 6]	**6 Marks**	**5 Marks**	**4 Marks**	**3 Marks**
		Completeness of code, consistent variable naming and formatting, well commented, uses existing skills in new ways/learns new skills to solve the experimental problem.	Completeness of code, consistent variable naming and formatting, lack of comments, uses existing skills in new ways/learns new skills to solve the experimental problem.	Completeness of code, inconsistent variable naming and formatting, lacks clarity in commenting, uses existing skills to solve the experimental problem.	Lack of completeness of code, improper variable naming and not formatted, lacks comments, uses existing skills to partially to solve the experimental problem.
2	Program Execution [Max. Marks 6]	**6 Marks**	**5 Marks**	**4-3 Marks**	**2-1 Marks**
		Program is free of errors and output is well formatted. Demonstrates	Program is free of errors and output is well formatted. Demonstrates	Program is free of errors and output is not properly	Program contains few logical errors and output is

		excellent problem solving and creativity skills.	better problem solving and creativity skills.	formatted. Demonstrates a clear understanding of the concepts relevant to the experiment.	not formatted. Demonstrates partial understanding of the concepts relevant to the experiment.
3	Program Testing and observation [Max. Marks 6]	6 Marks	5 Marks	4-3 Marks	2-1 Marks
		Able to give different inputs and record the output.	Able to give inputs and with help could record the output.	Able to give inputs and with help could record the output.	With help gives inputs and unable to record the output.
4	Lab report assessment [Max. Marks 6]	6 Marks	5 Marks	4-3 Marks	2-1 Marks
		Neatly written the record and regular to lab.	Neatly written the record, some data are missing in the record. Student is regular to lab.	Lack of neatness in the record, some data are missing in the record. Student is regular to lab..	Data are missing & not written the record neatly and irregular to lab
5	Viva [Max. Marks 6]	6 Marks	5 Marks	4-3 Marks	2-1 Marks
		Answers all the viva questions.	Answer 80% of the viva questions.	Answer 50% of the viva questions	Couldn't answer viva questions.

Table 3. Rubrics for Continuous Evaluation in Lab

Syllabus for the Academic Year – 2021 - 2022

Department: Computer Science and Engineering **Semester: VII**

Subject Name: ADVANCED DATABASE AND BIG DATA LAB (18CS706)

Subject Code: 18CS706 **L-T-P-C: 0-0-2-1**

Course Objectives:

Sl. No	Course Objectives
1	The course should enable the students to: Practice advance database programming concepts
2	Practice programming tools PIG and HIVE in Hadoop ecosystem.
3	Implement best practices for Hadoop development

Course Outcomes:

Course outcome	Descriptions
CO1	Implement the concept of Triggers, Cursors and procedures
CO2	Installation of VM-Ware and Hadoop Ecosystem
CO3	Implement the Piglatin scripts and HIVE programming

	CYCLE - I
1.	Write a program to implement database triggers in PL/SQL by using following schema - employee(e_id, e_name, e_doj, e_salary, e_age, primary key(e_id)) i. Create an employee table and insert any five records. ii. Write row-level trigger for salary changes and display the relevant message (insert / update / delete operations on employee)
2.	Write a program to implement database triggers in PL/SQL by using following schema – employee(e_id,e_name,e_age, primary key(e_id)) i. Create an employee table and insert any five records. ii. Write a trigger to check the age of an employee is between 18 to 58, if not raise an error. (during insert / update / delete operations on employee)
3.	Write a program to implement cursor in PL/SQL to display the employee details from the following table. emp (eno,ename,designation,doj,salary, primary key(eno))
4.	Write a program to implement Procedure in PL/SQL to update the salary of the employee from the following table. employee(eno, ename, designation, doj, salary , primary key(eno))
5.	Write a program to implement packages in PL/SQL by using following schema. employee(id, name, age, address, salary, primary key (id)); i. Create the package for adding, removing and listing a Employee. ii. Display suitable output.
	CYCLE- II
	• Installation of HADOOP in Ubuntu Form

- Installation of HIVE
- Installation of PIG

CYCLE- III

1. ***Execute the following commands in HADOOP***.

A. To get the list of directories and files at the root of HDFS.

B. To get the list of complete directories and files of HDFS.

C. To create a directory(say, sample) in HDFS.

D. To copy a file from local file system to HDFS.

E. To copy a file from HDFS to local file system

F. To copy a file from local file system to HDFS via copy FromLocal command

G. To copy a file from Hadoop file system to local file system via copy ToLocal command

H. To display the contents of an HDFS file on console

I. To copy a file from one directory to another directory

J. To remove a directory HDFS.

2. Execute the following commands in HIVE

A. To create a database named "STUDENTS" with comments and database properties

B. To display the list of all databases

C. To describe the database

D. To describe the extended database

E. To alter the database properties

F. To make the database as current working database

G. To drop database

H. To create managed table named 'STUDENT'.

I. To describe the "STUDENT" table

J. To create external table name "EXT_STUDENT".

K. To load data into the table from file named student.tsv

3. Execute the following commands in PIG

A. Find the tuples of those student where the GPA is greater than 4.0
Input : Student (rollno:int, name:chararray, gpa:float);

B. Display the name of all students in uppercase
Input : Student (rollno:int, name:chararray, gpa:float);

C. Group tuples of students based on gpa
Input : Student (rollno:int, name:chararray, gpa:float);

D. To remove duplicate tuples of students.
Input : Student (rollno:int, name:chararray, gpa:float);

E. Display the first 3 tuples from the "student" relation.
Input : Student (rollno:int, name:chararray, gpa:float);

F. Display the name of students in Ascending order
Input : Student (rollno:int, name:chararray, gpa:float);

G. To Join two relations namely "student" and "department " based on the values contained in the "rollno" column
Input : Student (rollno:int, name:chararray, gpa:float);
Department(rollno:int, deptno:int, deptname:chararray);

H. To merge the contents of relations namely "student" and "department "
Input : Student (rollno:int, name:chararray, gpa:float);
Department(rollno:int, deptno:int, deptname:chararray);

CO -PO MAPPING:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
CO1	3	3	2		2							
CO2	3	2			3							
CO3	3	3	3	3	3							2
CO4												

CYCLE I

1. Write a program to implement database triggers in PL/SQL by using following schema -employee(e_id,e_name,e_doj,e_salary,e_age)
   i. Create an employee table and insert any five records.
   ii. Write row-level trigger for salary changes.(insert / update / delete operations on employee)

```
CREATE TABLE EMPLOYEE(
E_ID NUMBER,
E_NAME VARCHAR(20),
E_DOJ VARCHAR(20),
```

```
E_SALARY NUMBER,
E_AGE NUMBER,
PRIMARY KEY (E_ID));

INSERT INTO EMPLOYEE VALUES(01,'THEJAS','2021-11-16',40000,21);
INSERT INTO EMPLOYEE VALUES(02,'SHREYAS','2022-01-01',50000,25);
INSERT INTO EMPLOYEE VALUES(03,'VIVEK','2022-02-01',50000,22);
INSERT INTO EMPLOYEE VALUES(04,'VINOD','2023-03-16',60000,24);
INSERT INTO EMPLOYEE VALUES(05,'VITTAL','2023-05-16',60000,23);

CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON employee
FOR EACH ROW
WHEN (NEW.E_ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.E_SALARY  - :OLD.E_SALARY;
dbms_output.put_line('Old salary: ' || :OLD.E_SALARY);
dbms_output.put_line('New salary: ' || :NEW.E_SALARY);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
SET SERVEROUTPUT ON;

INSERT INTO EMPLOYEEVALUES(06,'SINDHU','2024-06-03',75000,25);

UPDATE EMPLOYEE
```

```
SET E_SALARY=E_SALARY+500
WHERE E_ID=2;

DELETE EMPLOYEE
WHERE E_ID=2;
```

2. Write a program to implement database triggers in PL/SQL by using following schema – employee2(e_id,e_name,e_age)
   iii. Create an employee table and insert any five records.
   iv. Write a trigger to check the age of an employee is between 18 to 58, if not raise an error.(during insert / update / delete operations on employee)

```
CREATE TABLE EEMP2(
E_ID NUMBER,
E_NAME VARCHAR(20),
E_AGE NUMBER,
PRIMARY KEY (E_ID));

INSERT INTO EEMP2 VALUES(01,'THEJAS',21);
INSERT INTO EEMP2 VALUES(02,'SHREYAS',25);
INSERT INTO EEMP2 VALUES(03,'VIVEK',22);
INSERT INTO EEMP2 VALUES(04,'VINOD',24);
INSERT INTO EEMP2 VALUES(12,'VITTAL',11);

CREATE OR REPLACE TRIGGER DISPLAY_AGE_CHANGES
BEFORE INSERT OR UPDATE OR DELETE ON EEMP2
FOR EACH ROW
WHEN(NEW.E_ID>0)
BEGIN
IF:NEW.E_AGE < 18
THEN
```

```
RAISE_APPLICATION_ERROR(-20001,'Employee age must be greater than or
equal to 18.');

ELSIF:NEW.E_AGE > 58

THEN

RAISE_APPLICATION_ERROR(-20001,'Employee age must be lesser than or equal
to 58.');

END IF;

END;

/

SET SERVEROUTPUT ON;

INSERT INTO eemp2 (12,'VITTAL', 11);

UPDATE eemp2

SET E_age=E_age+10

WHERE E_ID=12;

DELETE EEMP2

WHERE E_ID=2;
```

3. Write a program to implement cursor in PL/SQL to display the employee details from the following table -emp(eno,ename,designation,doj,salary) .

```
CREATE TABLE EMP3(
E_ID NUMBER,
E_NAME VARCHAR(20),
SALARY NUMBER,
PRIMARY KEY (E_ID));


INSERT INTO EMP3 VALUES(01,'NATASHA',35000);
INSERT INTO EMP3 VALUES(02,'STEVE',40000);
INSERT INTO EMP3 VALUES(03,'STARK',50000);
INSERT INTO EMP3 VALUES(04,'CLINT',35000);
```

```
INSERT INTO EMP3 VALUES(05,'PAUL',30000);
INSERT INTO EMP3 VALUES(06,'MARK',15000);

CREATE TABLE emp_temp AS
 SELECT * FROM   emp3;

DECLARE
   CURSOR employee_cur IS
    SELECT * FROM   emp3
    FOR UPDATE;
incr_sal NUMBER;
BEGIN
   FOR employee_rec IN employee_cur LOOP
      IF employee_rec.salary< 35000 THEN
incr_sal := .20;
      ELSE
incr_sal := .10;
      END IF;

      UPDATE emp3
      SET    salary = salary + salary * incr_sal
WHERE  CURRENT OF employee_cur;
   END LOOP;
END;
/


CREATE TABLE EMP5(

E_ID NUMBER,

E_NAME VARCHAR(20),

SALARY NUMBER,

PRIMARY KEY (E_ID));


INSERT INTO EMP5 VALUES(100,'ROCK',50000);

INSERT INTO EMP5 VALUES(97,'BIG SHAW',14000);

INSERT INTO EMP5 VALUES(150,'HHH',5000);
```

```
INSERT INTO EMP5 VALUES(140,'ROMAN REIGNS',35000);
INSERT INTO EMP5 VALUES(143,'JOHN CENA',30000);
INSERT INTO EMP5 VALUES(80,'UNDER TAKER',24000);
```

4. Write a program to implement Procedure in PL/SQL to update the salary of the employee from the following table – employee5(eno,ename,designation,doj,salary) .

```
CREATE OR REPLACE PROCEDURE adjust_salary
IS
BEGIN
UPDATE EMP5 set salary = salary * 1.1 WHERE salary>25000;
UPDATE EMP5 set salary = salary * 1.2 WHERE salary<25000;
END;
/
```

PL/SQL procedure successfully completed.

```
Exec adjust_salary;
```

5. Write a program to implement packages in PL/SQL by using following schema – EMPLOYEE12(ID NUMBER,NAME VARCHAR(20),AGE NUMBER,ADDRESS VARCHAR(20),SALARY NUMBER, PRIMARY KEY (ID));
   iii. Create the package for adding, removing and listing a customer.
   iv. Display suitable output.

Package:

```
CREATE TABLE EMPLOYEE12(
ID NUMBER,
NAME VARCHAR(20),
```

```
AGE NUMBER,
ADDRESS VARCHAR(20),
SALARY NUMBER,
PRIMARY KEY (ID));

INSERT INTO EMPLOYEE12 VALUES(01,'THEJAS',21,'CHELUR',40000);
INSERT INTO EMPLOYEE12 VALUES(02,'SHREYAS',23,'MUDIGERE',50000);
INSERT INTO EMPLOYEE12 VALUES(03,'VIVEK',25,'HAROGERI',40000);
INSERT INTO EMPLOYEE12 VALUES(04,'SINDHU',21,'SOGILU',35000);

create or replace package e_pack as
Procedure add emp(e_id employee12.id%type,
e_name employee12.name%type,
e_age employee12.age%type,
e_addr employee12.address%type,
e_sal employee12.salary%type);

Procedure del emp(e_id employee12.id%type);
Procedure list emp;
End e_pack;
  /
//package created//

create or replace package body e_pack as
Procedure add emp(e_id employee12.id%type,
e_name employee12.name%type,
e_age employee12.age%type,
e_addr employee12.address%type,
```

```
e_sal employee12.salary%type)
is
begin
insert into employee12(id, name,age,address,salary)
values(e_id, e_name,e_age,e_addr,e_sal);
endaddemp;

Procedure del emp(e_id employee12.id%type) is
begin
delete from employee12 where id=e_id;
End del emp;
Procedure list emp is
Cursor e_emp is
select name from employee12;
TYPE e_list is table of employee12.name%type;
name_liste_list := e_list();
counter integer := 0;
begin
for n in e_emp loop
counter := counter +1;
name_list.extend;
name_list(counter) := n.name;
dbms_output.put_line('employee('||counter||')'||name_list(counter));
end loop;
endlistemp;
ende_pack;
/
:////Package body created.///:
```

```
Set Serveroutput on;

declare
code employee12.id%type:=10;
begin
e_pack.add emp(12, 'shrey',39,'tumkur',7600);
e_pack.lis temp;
e_pack.del emp(code);
e_pack.list emp;
end;
/
PL/SQL procedure successfully completed.
```

CYCLE II

Installation of Hadoop

- **Hardware Requirement:** Desktop Computer / laptop computer.
- **Software Requirement:** UBUNTU 14.0 Operating System with HADOOP Installed
  PL/SQL development kit for Part A

Hadoop software can be installed in three modes ofoperation:

- **Stand Alone Mode**: Hadoop is a distributed software and is designed to run on a commodity of machines. However, we can install it on a single node in stand-alone mode. In this mode, Hadoop software runs as a single monolithic java process. This mode is extremely useful for debugging purpose. You can first testrun your Map-Reduce application in this mode on small data, before actually executing it on cluster with big data.
- **Pseudo Distributed Mode**: In this mode also, Hadoop software is installed on a Single Node.Various daemons of Hadoop will run on the same machine as separate java processes. Hence all the daemons namely NameNode, DataNode, SecondaryNameNode, JobTracker, TaskTracker run on single machine.
- **Fully Distributed Mode**: In Fully Distributed Mode, the daemons NameNode, JobTracker, SecondaryNameNode (Optional and can be runon a separate node) run on the Master Node. The daemons DataNode and TaskTracker runon the Slave Node.

Hadoop Installation: Ubuntu Operating System in stand-alonemode

Steps for Installation

Prerequisites

First, we need to make sure that the following prerequisites are installed:

1. Java 8 runtime environment (JRE): Hadoop 3 requires a Java 8 installation. Prefer using the offline installer.
2. Java 8 development Kit (JDK)
3. To unzip downloaded Hadoop binaries, we should install 7zip.
4. Create a folder "E:\hadoop-env" on my local machine to store downloaded files.

2. Download Hadoop binaries

The first step is to download Hadoop binaries from the official website. The binary package size is about 342 MB.

After finishing the file download, we should unpack the package using 7zip int two steps. First, we should extract the hadoop-3.2.1.tar.gz library, and then, we should unpack the extracted tar file:

Extracting hadoop-3.2.1.tar.gz package using 7zip

Extracted hadoop-3.2.1.tar file

The tar file extraction may take some minutes to finish. In the end, you may see some warnings about symbolic link creation. Just ignore these warnings since they are not related to windows.

After unpacking the package, we should add the Hadoop native IO libraries, which can be found in the following GitHub repository: https://github.com/cdarlint/winutils.

Since we are installing Hadoop 3.2.1, we should download the files located in https://github.com/cdarlint/winutils/tree/master/hadoop-3.2.1/bin and copy them into the "hadoop-3.2.1\bin" directory.

3. Setting up environment variables

After installing Hadoop and its prerequisites, we should configure the environment variables to define Hadoop and Java default paths.

To edit environment variables, go to Control Panel > System and Security > System (or right-click > properties on My Computer icon) and click on the "Advanced system settings" link.

Opening advanced system settings

When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.

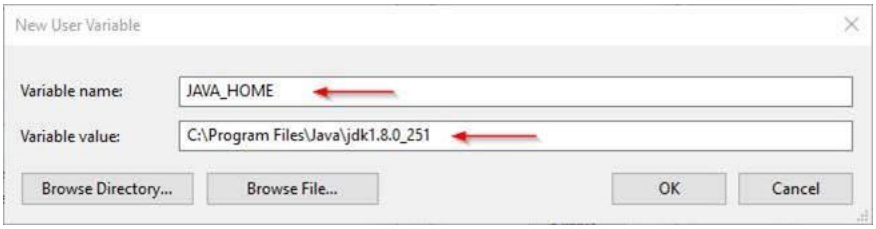
Advanced system settings dialog

In the “Environment Variables” dialog, press the “New” button to add a new variable.

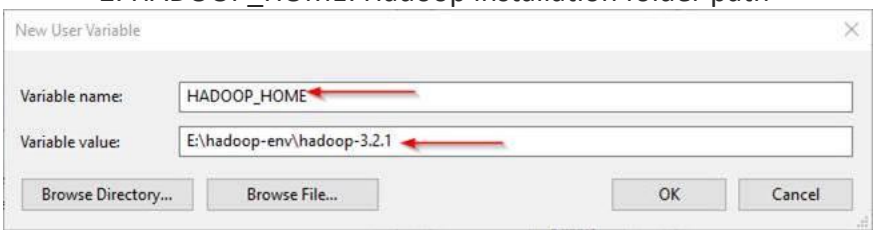
Note: In this guide, we will add user variables since we are configuring Hadoop for a single user. If you are looking to configure Hadoop for multiple users, you can define System variables instead.

There are two variables to define:

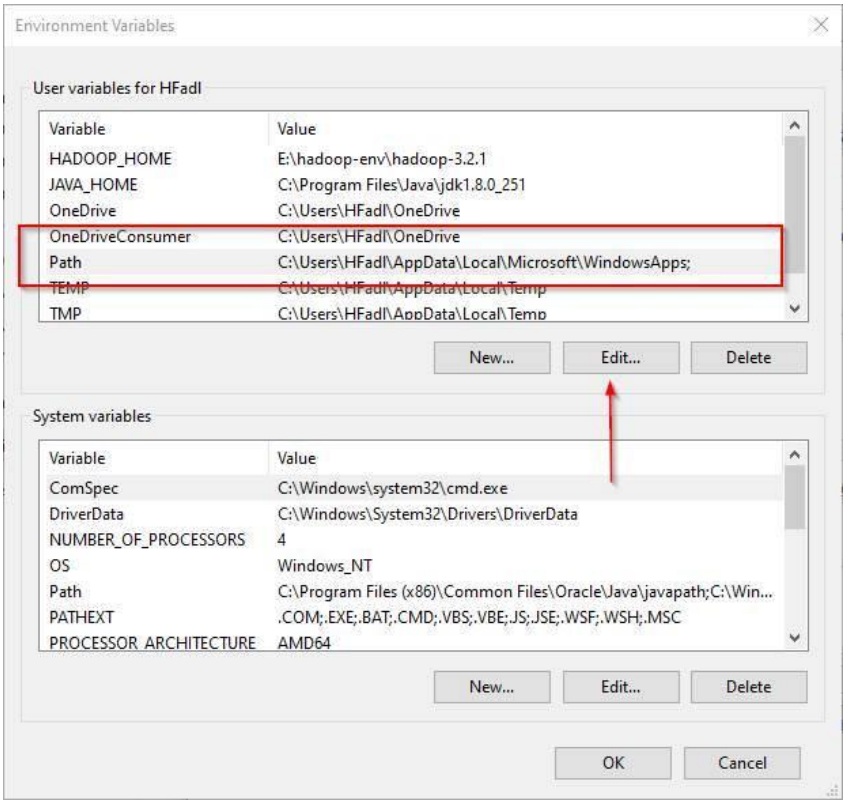
1. JAVA_HOME: JDK installation folder path



2. HADOOP_HOME: Hadoop installation folder path



Editing PATH variable



3.1. JAVA_HOME is incorrectly set error

Now, let's open PowerShell and try to run the following command:

hadoop -version

In this example, since the JAVA_HOME path contains spaces, I received the following error:

JAVA_HOME is incorrectly set

To solve this issue, we should use the windows 8.3 path instead. As an example:

- Use "Progra~1" instead of "Program Files"
- Use "Progra~2" instead of "Program Files(x86)"

After replacing "Program Files" with "Progra~1", we closed and reopened PowerShell and tried the same command. As shown in the screenshot below, it runs without errors.

4. Configuring Hadoop cluster

There are four files we should alter to configure Hadoop cluster:

1. %HADOOP_HOME%\etc\hadoop\hdfs-site.xml
2. %HADOOP_HOME%\etc\hadoop\core-site.xml
3. %HADOOP_HOME%\etc\hadoop\mapred-site.xml
4. %HADOOP_HOME%\etc\hadoop\yarn-site.xml

4.1. HDFS site configuration

As we know, Hadoop is built using a master-slave paradigm. Before altering the HDFS configuration file, we should create a directory to store all master node (name node) data and another one to store data (data node). In this example, we created the following directories:

- E:\hadoop-env\hadoop-3.2.1\data\dfs\namenode
- E:\hadoop-env\hadoop-3.2.1\data\dfs\datanode

Now, let's open "hdfs-site.xml" file located in "%HADOOP_HOME%\etc\hadoop" directory, and we should add the following properties within the <configuration></configuration> element:

<property><name>dfs.replication</name><value>1</value></property><property><name>dfs.namenode.name.dir</name><value>file:///E:/hadoop-env/hadoop-3.2.1/data/dfs/namenode</value></property><property><name>dfs.datanode.data.dir</name><value>file:///E:/hadoop-env/hadoop-3.2.1/data/dfs/datanode</value></property>

Note that we have set the replication factor to 1 since we are creating a single node cluster.

4.2. Core site configuration

Now, we should configure the name node URL adding the following XML code into the <configuration></configuration> element within "core-site.xml":

<property><name>fs.default.name</name><value>hdfs://localhost:9820</value></property>

4.3. Map Reduce site configuration

Now, we should add the following XML code into the <configuration></configuration> element within "mapred-site.xml":

<property><name>mapreduce.framework.name</name><value>yarn</value><description>MapReduce framework name</description></property>

4.4. Yarn site configuration

Now, we should add the following XML code into the <configuration></configuration> element within "yarn-site.xml":

<property><name>yarn.nodemanager.aux-services</name><value>mapreduce_shuffle</value><description>Yarn Node Manager Aux Service</description></property>

5. Formatting Name node

After finishing the configuration, let's try to format the name node using the following command:

hdfs namenode -format

Due to a bug in the Hadoop 3.2.1 release, you will receive the following error:

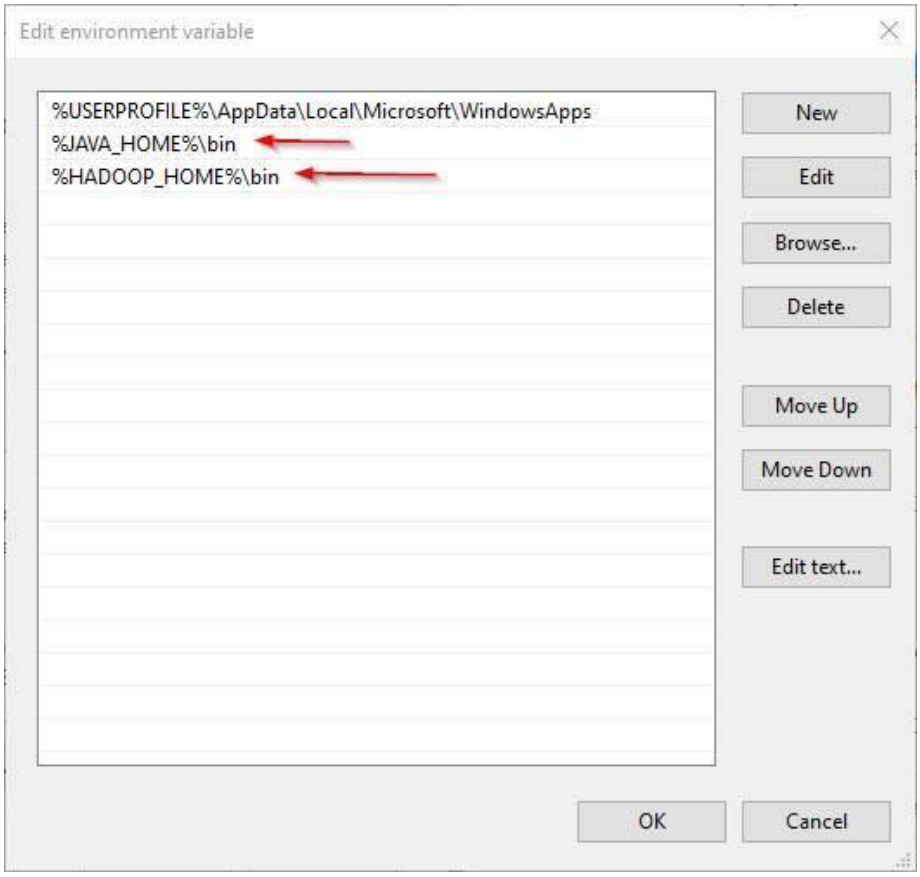
2020–04–17 22:04:01,503 ERROR namenode.NameNode: Failed to start namenode.java.lang.UnsupportedOperationExceptionat java.nio.file.Files.setPosixFilePermissions(Files.java:2044)at org.apache.hadoop.hdfs.server.common.Storage$StorageDirectory.clearDirectory(Storage.java:452)at org.apache.hadoop.hdfs.server.namenode.NNStorage.format(NNStorage.java:591)at org.apache.hadoop.hdfs.server.namenode.NNStorage.format(NNStorage.java:613)at org.apache.hadoop.hdfs.server.namenode.FSImage.format(FSImage.java:188)at org.apache.hadoop.hdfs.server.namenode.NameNode.format(NameNode.java:1206)at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1649)at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1759)2020–04–17 22:04:01,511 INFO util.ExitUtil: Exiting with status 1: java.lang.UnsupportedOperationException2020–04–17 22:04:01,518 INFO namenode.NameNode: SHUTDOWN_MSG:

This issue will be solved within the next release. For now, you can fix it temporarily using the following steps (reference):

1. Download hadoop-hdfs-3.2.1.jar file from the following link.
2. Rename the file name hadoop-hdfs-3.2.1.jar to hadoop-hdfs-3.2.1.bak in folder %HADOOP_HOME%\share\hadoop\hdfs
3. Copy the downloaded hadoop-hdfs-3.2.1.jar to folder %HADOOP_HOME%\share\hadoop\hdfs

Now, if we try to re-execute the format command (Run the command prompt or PowerShell as administrator), you need to approve file system format.

And the command is executed successfully:



6. Starting Hadoop services

Now, we will open PowerShell, and navigate to "%HADOOP_HOME%\sbin" directory. Then we will run the following command to start the Hadoop nodes:

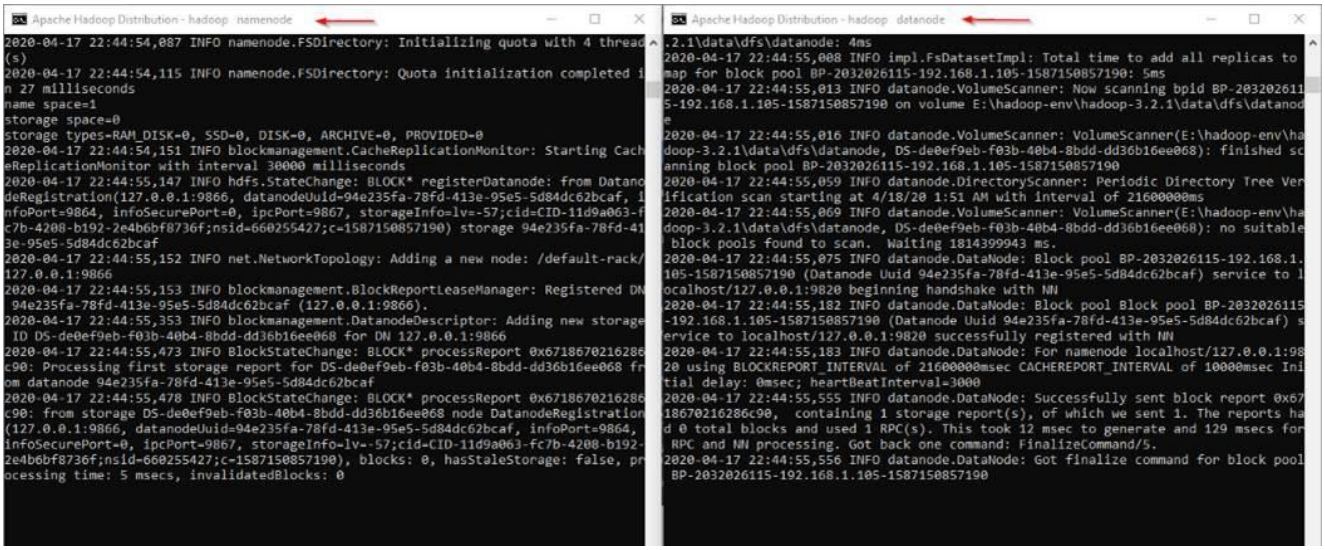
.\start-dfs.cmd

```
2020-04-17 22:14:17,206 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2032026115-192.168.1.105-1587150857190
2020-04-17 22:14:17,207 INFO common.Storage: Will remove files: []
2020-04-17 22:14:17,275 INFO common.Storage: Storage directory E:\hadoop-env\hadoop-3.2.1\data\dfs\namenode has been suc
cessfully formatted.
2020-04-17 22:14:17,331 INFO namenode.FSImageFormatProtobuf: Saving image file E:\hadoop-env\hadoop-3.2.1\data\dfs\namen
ode\current\fsimage.ckpt_0000000000000000000 using no compression
2020-04-17 22:14:17,531 INFO namenode.FSImageFormatProtobuf: Image file E:\hadoop-env\hadoop-3.2.1\data\dfs\namenode\cur
rent\fsimage.ckpt_0000000000000000000 of size 400 bytes saved in 0 seconds .
2020-04-17 22:14:17,555 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2020-04-17 22:14:17,580 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2020-04-17 22:14:17,580 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at [redacted]
************************************************************/
PS C:\Windows\system32>
```

Two command prompt windows will open (one for the name node and one for the data node) as follows:

Next, we must start the Hadoop Yarn service using the following command:

./start-yarn.cmd




```
PS E:\hadoop-env\hadoop-3.2.1\sbin> jps
14560 DataNode
4960 ResourceManager
5936 NameNode
768 NodeManager
14636 Jps
PS E:\hadoop-env\hadoop-3.2.1\sbin>
```

To make sure that all services started successfully, we can run the following command:

jps

It should display the following services:

14560 DataNode

4960 ResourceManager

5936 NameNode

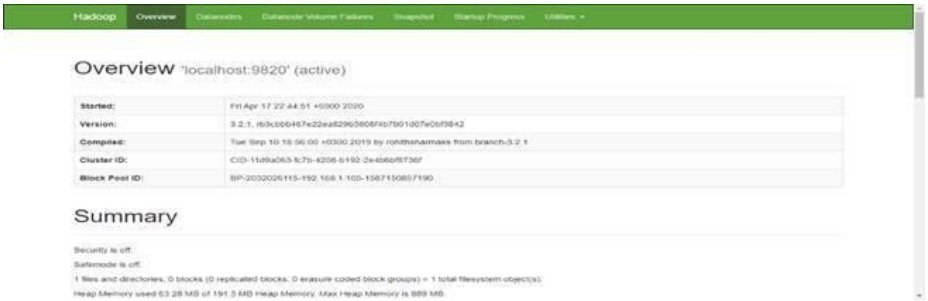
768 NodeManager

14636 Jps

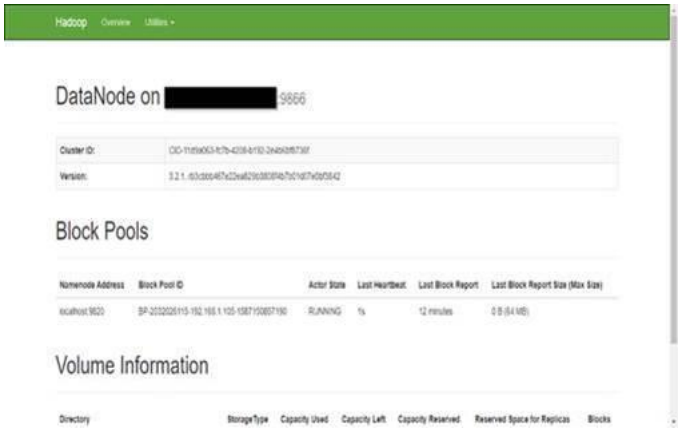
7. Hadoop Web UI

There are three web user interfaces to be used:

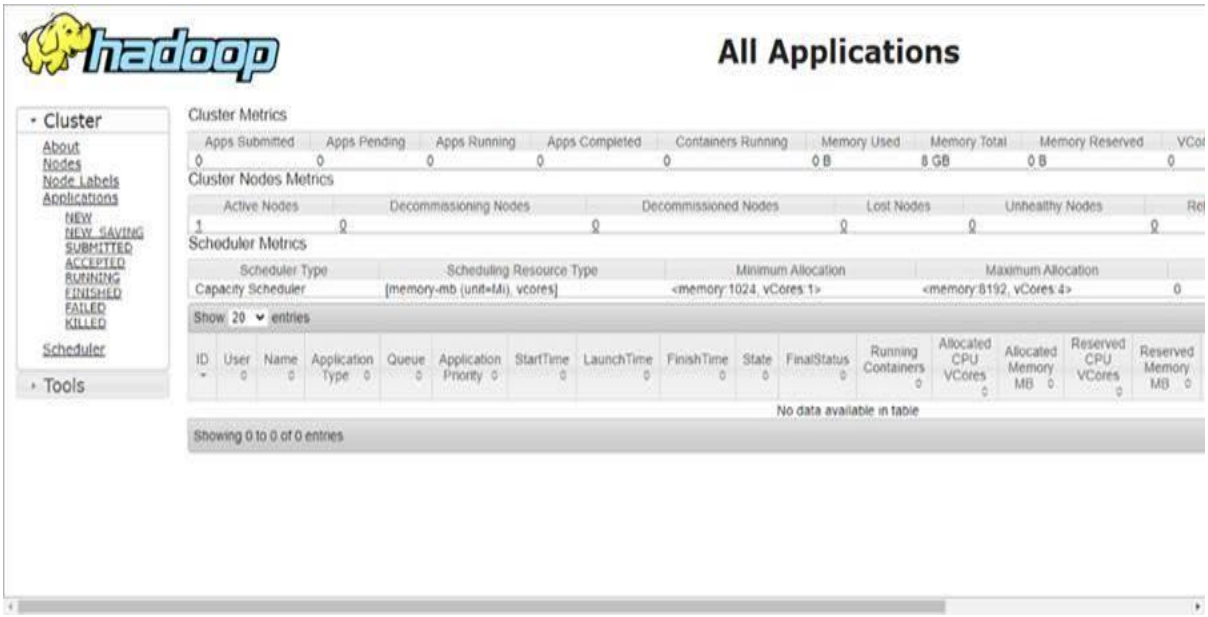
- Name node web page: http://localhost:9870/dfshealth.html



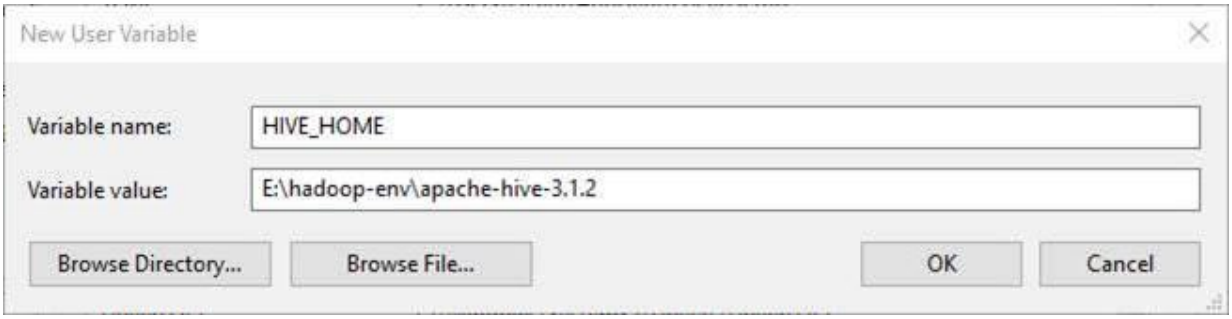
- Data node web page: http://localhost:9864/datanode.html

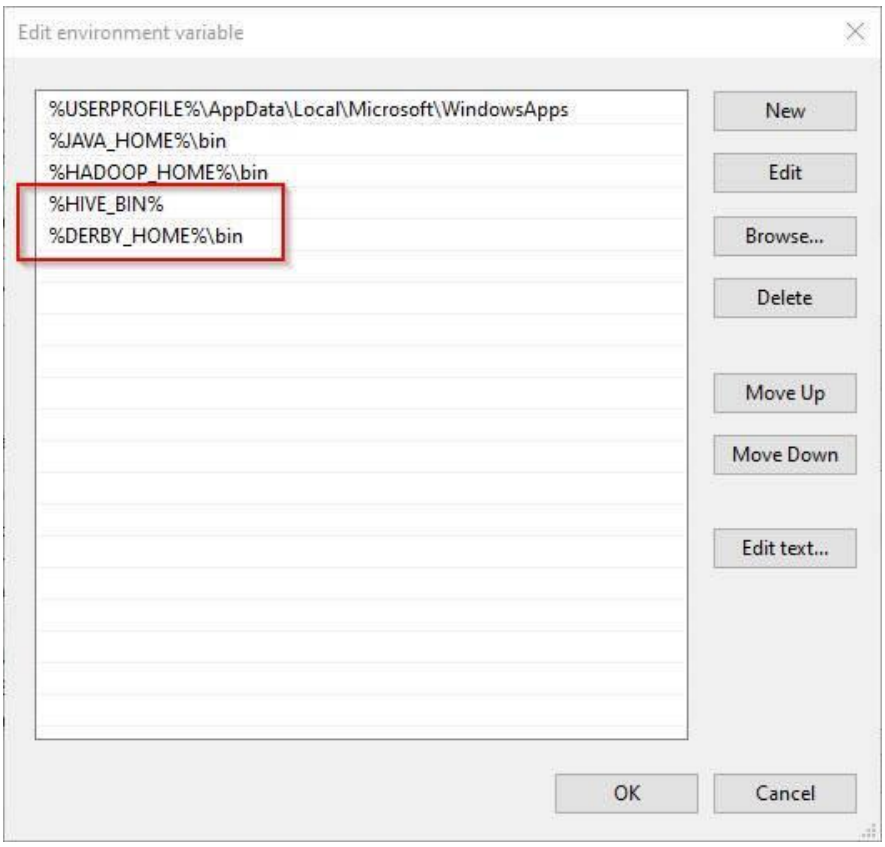


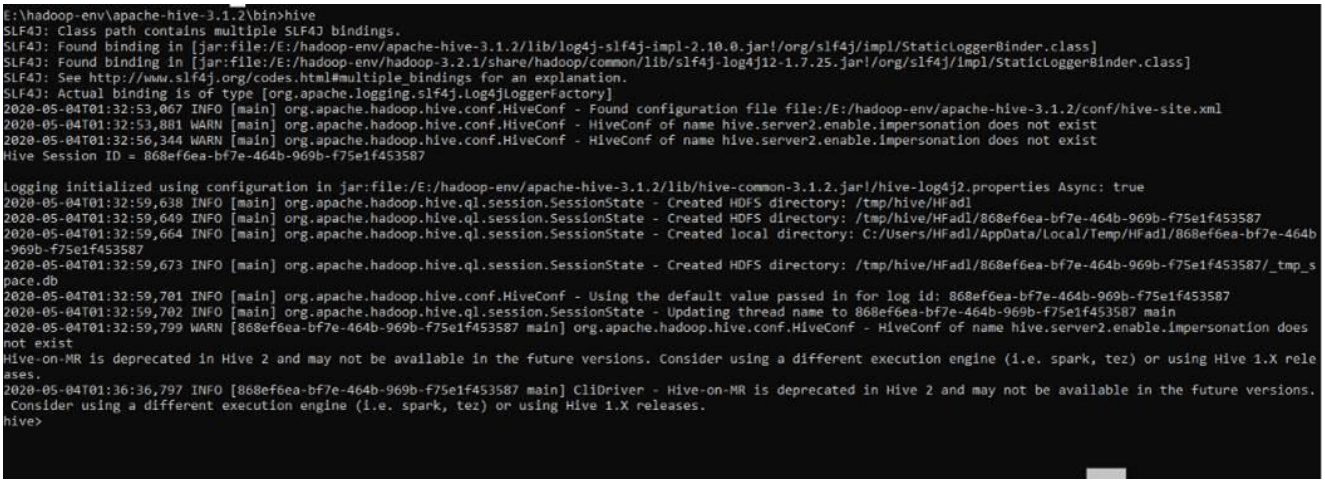
- Yarn web page: http://localhost:8088/cluster



HIVE installation:







After installing Hadoop, you should install Apache Pig.

1. Downloading Apache Pig

To download the Apache Pig, you should go to the following link:

- https://downloads.apache.org/pig/

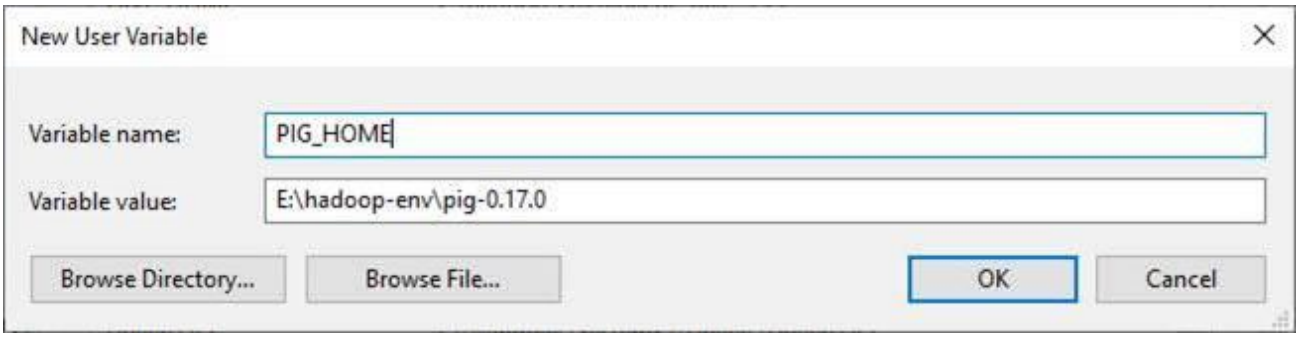
If you are looking for the latest version, navigate to “latest” directory, then download the pig-x.xx.x.tar.gz file.

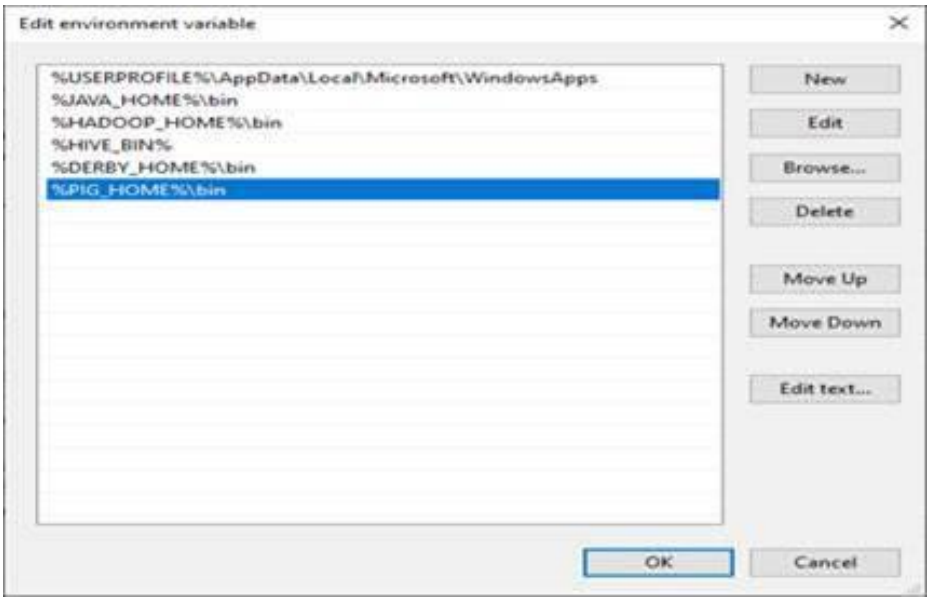
After the file is downloaded, we should extract it twice using 7zip *(using 7zip: the first time we extract the .tar.gz file, the second time we extract the .tar file).* We will extract the Pig folder into "E:\hadoop-env" directory as used in the previous articles.

2. Setting Environment Variables

After extracting Derby and Hive archives, we should go to Control Panel > System and Security > System. Then Click on "Advanced system settings".

In the advanced system settings dialog, click on "Environment variables" button.





3. Starting Apache Pig

After setting environment variables, let's try to run Apache Pig.

***Note**: Hadoop Services must be running*

Open a command prompt as administrator, and execute the following command

pig -version

You will receive the following exception:

'E:\hadoop-env\hadoop-3.2.1\bin\hadoop-config.cmd' is not recognized as an internal or external command,

operable program or batch file.

'-Xmx1000M' is not recognized as an internal or external command,

operable program or batch file.

```
E:\>pig -version
'E:\hadoop-env\hadoop-3.2.1\bin\hadoop-config.cmd' is not recognized as an internal or external command,
operable program or batch file.
'-Xmx1000M' is not recognized as an internal or external command,
operable program or batch file.
```

To fix this error, we should edit the pig.cmd file located in the "pig-0.17.0\bin" directory by changing the HADOOP_BIN_PATH value from "%HADOOP_HOME%\bin" to "%HADOOP_HOME%\libexec". Now, let's try to run the "pig -version" command again:

```
E:\>pig -version
Apache Pig version 0.17.0 (r1797386)
compiled Jun 02 2017, 15:41:58
```

The simplest way to write PigLatin statements is using Grunt shell which is an interactive tool where we write a statement and get the desired output. There are two modes to involve Grunt Shell:

1. Local: All scripts are executed on a single machine without requiring Hadoop. (command: pig -x local)
2. MapReduce: Scripts are executed on a Hadoop cluster (command: pig -x MapReduce)

Since we have installed Apache Hadoop 3.2.1 which is not compatible with Pig 0.17.0, we will try to run Pig using local mode.

```
E:\>pig -x local
2020-05-05 03:22:24,894 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2020-05-05 03:22:24,895 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2020-05-05 03:22:25,246 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02
2020-05-05 03:22:25,246 [main] INFO  org.apache.pig.Main - Logging error messages to: E:\hadoop-env\hadoop-3.2.
2020-05-05 03:22:25,282 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file C:\Users\HFadl/.pigbo
2020-05-05 03:22:25,495 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is d
e.jobtracker.address
2020-05-05 03:22:25,501 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connectin
:///
2020-05-05 03:22:25,912 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum i
ytes-per-checksum
2020-05-05 03:22:25,960 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-4a3a

2020-05-05 03:22:25,962 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.ena
grunt>
```

References

1. Apache Pig official website
2. SolutionMandi: Pig Installation on Windows 10

CYCLE III

1. Basic Commands: Working with HDFS Commands

To get the list of directories and files at the root of HDFS.

Action: hadoop fx –ls/

To get the list of complete directories and files of HDFS.

Action: hadoop fs –ls –R/

To create a directory(say, sample) in HDFS.

Action: hadoop fs –mkdir / sample

To copy a file from local file system to HDFS.

Action: hadoop fs –put/root/sample/test.txt /sample/test.txt

To copy a file from HDFS to local file system

Action: hadoop fs –get/sample/test.txt /root/sample/testsample.txt

To copy a file from local file system to HDFS via copy FromLocal command

Action: hadoop fs –copyFromLocal /root/sample/test.txt /sample/testsample.txt

To copy a file from Hadoop file system to local file system via copy ToLocal command

Action: hadoop fs –copytoLocal /sample/test.txt /root/sample/testsample1.txt

To display the contents of an HDFS file on console

Action: hadoop fs –cat/sample/test.txt

To copy a file from one directory to another directory

Action: hadoop fs –cp/sample/test.txt /sample1

To remove a directory HDFS.

Action: hadoop fs-rm-r/sample1

1. Basic Commands: Working with HIVE Commands

To create a database named “STUDENTS” with comments and database properties

Action: CREATE DATABASE IF NOT EXISTS STUDENTS COMMENT ‘STUDENT Details’ with DB properties(‘creator’ = ‘SSIT’);

To display a list of all databases

Action: SHOW DATABASES;

To describe a database

Action: DESCRIBE DATABASE STUDENTS;

Note: show only DB name, comment and DB directory

To describe the extended database

Action: DESCRIBE DATABASE EXTENDED STUDENTS;

Note: shows DB properties also

To alter the database properties

Action: ALTER DATABASE STUDENTS SET DBPROPERTIES(‘EDITED-BY’ = ‘CSE’);

Note: in Hive, it is not possible to unser to DB properties

To make the database as current working database

Action: USE STUDENTS;

To drop database

Action: DROP DATABASE STUDENTS;

Note: Hive creates database in warehouse directory of Hive

Managed table

To create managed table named ‘STUDENT’.

Action: CREATE TALBE IF NOT EXISTS STUDENT(rollno INT, name STRING, gpa FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘\t’);

To describe the “STUDENT” table

Action: DESCRIBE STUDENT;

Note: Hive creates database in warehouse directory of Hive

To create external table name "EXT_STUDENT".

Action: CREATE EXTERNAL TABLE IF NOT EXISTS EXT_STUDENT(rollno INT, name STRING, gpa FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION 'STUDENT_INFO;

Note: Hive creates external table in the specified location

Loading of data into table from file

To load data into the table from file named student.tsv

Action: LOAD DATA LOCAL INPATH '/root/hivedemos/student.tsv' OVERWRITE INTO TABLE EXT_STUDENT;

Note: local keyword is used to load the data from the local file system, to load the data from HDFS remove local key word from statement

Basic Commands: Working with PIG Commands

1. We can run Pig in two ways
2. Interactive Mode
3. Batch mode

Interactive Mode:

Pig in interactive mode by invoking GRUNT shell.

Local mode

Pig in local mode, you need to have your files in the local file system.

Action: pig –r local filename

MapReduce mode

Action: pig filename

HDFS commands

RELATIONAL OPERATORS -FILTER

Find the tuples of those student where the GPA is greater than 4.0

Input : Student (rollno:int, name:chararray, gpa:float);

Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=filter A by gpa > 4;
DUMP B;

RELATIONAL OPERATORS –FOREACH

Display the name of all students in uppercase

Input : Student (rollno:int, name:chararray, gpa:float);

Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=foreach A generate UPPER(name);
DUMP B;

RELATIONAL OPERATORS –GROUP

Group tuples of students based on gpa

Input : Student (rollno:int, name:chararray, gpa:float);

Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=GROUP A BY GPA;
DUMP B;

RELATIONAL OPERATORS –DISTINCT

To remove duplicate tuples of students.

Input : Student (rollno:int, name:chararray, gpa:float);

Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=DISTINCT A;
DUMP B;

RELATIONAL OPERATORS –LIMIT

Display the first 3 tuples from the "student" relation.

Input : Student (rollno:int, name:chararray, gpa:float);

Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=LIMIT A 3;
DUMP B;

RELATIONAL OPERATORS –ORDER BY

Display the name of students in Ascending order

Input : Student (rollno:int, name:chararray, gpa:float);
Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B=ORDER A BY name;
DUMP B;

RELATIONAL OPERATORS –JOIN

To Join two relations namely "student" and "department " based on the values contained in the "rollno" column
Input : Student (rollno:int, name:chararray, gpa:float);
Department(rollno:int, deptno:int, deptname:chararray);
Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B= load '/pigdemo/student.tsv' as (rollno:int, deptno:int, deptname:chararray);
C= JOIN A BY rollno, B BY rollno;
DUMP C;
DUMP B;

RELATIONAL OPERATORS –UNION

To merge the contents of relations namely "student" and "department "
Input : Student (rollno:int, name:chararray, gpa:float);
Department(rollno:int, deptno:int, deptname:chararray);
Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
B= load '/pigdemo/student.tsv' as (rollno:int, deptno:int, deptname:chararray);
C= UNION A, B;
STORE C INTO '/pigdemo/uniondemo';
DUMP B;

RELATIONAL OPERATORS –SPLIT

To partition a relation based on the GPAs acquired by the students.
*GPA =4.0, place it into relation X,
*GPA is <4.0 place it into relation Y.
Input : Student (rollno:int, name:chararray, gpa:float);
Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
SPLIT A into X if gpa==4.0, Y if GPA<=4.0;
DUMP X;

RELATIONAL OPERATORS –SAMPLE

To depict the use of sample
Input : Student (rollno:int, name:chararray, gpa:float);
Action: A=load '/pigdemo/student.tsv' as (rollno:int, name:chararray, gpa:float);
Sample A 0.01
DUMP B;

EVAL FUNCTION –AVG

To calculate the average marks for each student.
Input : Student (studname:chararray, marks:int);
Action: A=load '/pigdemo/student.tsv' as (studname:chararray, marks:int);
B= GROUP A BY studname;
C=FOREACH B GENERATE A.studname, AVG(A.marks);
DUMP C;

EVAL FUNCTION –MAX

To calculate the maximum marks for each student.
Input : Student (studname:chararray, marks:int);
Action: A=load '/pigdemo/student.tsv' as (studname:chararray, marks:int);
B= GROUP A BY studname;
C=FOREACH B GENERATE A.studname, MAX(A.marks);
DUMP C;

EVAL FUNCTION –COUNT

To count the number of tuples in bag
Input : Student (studname:chararray, marks:int);
Action: A=load '/pigdemo/student.tsv' USING PigStore(',')as (studname:chararray, marks:int);
B= GROUP A BY studname;
C=FOREACH B GENERATE A.studname, COUNT(A);
DUMP C;