LAMBTON COLLEGE

A picture containing text

Description automatically generated

A Project on

[Hadoop Ecosystem]

121 Brunel Rd, Mississauga

ON L4Z 3E9

A Group project for analyzing stack overflow data in Hive on Top of Hadoop

Big Data Analytics DSMM

Under the supervision

of

Professor Teresa Zhu

**Submitted BY:Group E**

Aadarsha Chapagain (C0825975)

Onyinye Mbanefo (C0831578)

Roshan Acharya (C0831342)

Anjana Kuriakose (C0829580)

**Submitted To:**

Lambton College

Professor Teresa Zhu

**Submission Date:**

27th September 2022

# Step1:

## 1. Download the zip file from the following location & move it to your Hadoop environment. Place this data under /LDZ/data/ in Hadoop

### Architecture for Hive

Diagram

Description automatically generated

### Hadoop on Cloud Environment

Ubuntu 18.04 was installed on AWS ec2(**t2.large**) with 8gb of memory and 300gb of storage. Hadoop 2.6, Hive and mysql was installed on the machine.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Single Node on Hadoop was set up using following commands on AWS ec2.

1. create a new 'hadoop' user in ubuntu.

2. create '/home/hadoop/work' and '/home/hadoop/work/hadoopdata' folders

mkdir /home/hadoop/work

mkdir /home/hadoop/work/hadoopdata

3. Download 'hadoop-2.6.0.tar.gz' file from Apache mirrors https://archive.apache.org/dist/hadoop/core/hadoop-2.6.0/, copy 'hadoop-2.6.0.tar.gz' file into this '/home/hadoop/work' directory and extract the tar file into same directory.

tar -xvzf hadoop-2.6.0.tar.gz

4. Open the '~/.bashrc' file on all the machines and add the following lines at the end and save:

command: gedit ~/.bashrc

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export HADOOP\_HOME=/home/hadoop/work/hadoop-2.6.0

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export PATH=$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

5. Enter the below commands on terminal:

ssh localhost

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

6. Update the '/home/hadoop/work/hadoop-2.6.0/etc' folder files 'hadoop-env.sh','core-site.xml','hdfs-site.xml','mapred-site.xml', 'yarn-env.sh', 'yarn-site.xml','masters' and 'slaves' files as per the below configurations

hadoop-env.sh

=============

# The java implementation to use.

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export JAVA\_HOME=${JAVA\_HOME}

core-site.xml

=============

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:8020</value>

</property>

</configuration>

hdfs-site.xml

=============

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/hadoop/work/hadoopdata/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/hadoop/work/hadoopdata/dfs/data</value>

</property>

<property>

<name>dfs.namenode.checkpoint.dir</name>

<value>file:/home/hadoop/work/hadoopdata/dfs/namesecondary</value>

</property>

</configuration>

mapred-env.sh

=============

# export JAVA\_HOME=/home/y/libexec/jdk1.8.0/

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

mapred-site.xml

===============

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

yarn-env.sh

===========

# export JAVA\_HOME=/home/y/libexec/jdk1.8.0/

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

yarn-site.xml

=============

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>localhost</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce\_shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

slaves

======

localhost

7. Format the 'namenode' from current machine using this command:

hadoop namenode -format or hdfs namenode -format

8. Start the hadoop by using this command on current machine:

start-dfs.sh

start-all.sh (depricated)

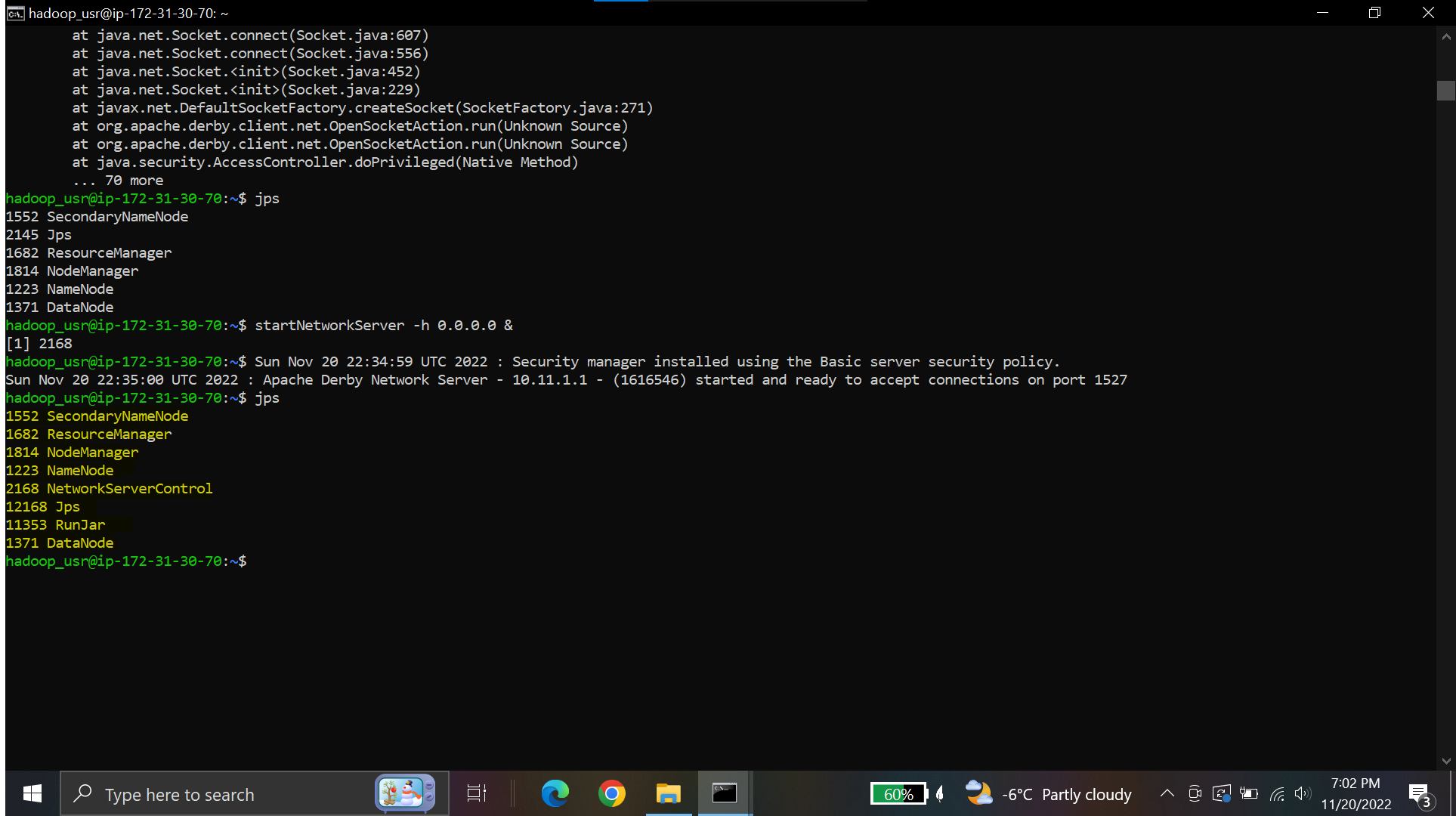
9. Stop the hadoop by using this command on current machine:

start-yarn.sh

stop-all.sh (depricated)

10. jps

With JPS it can been seen that namenode and resource manager are started.



To check hadoop storage hdfs dfs -df -h command can be used

A screenshot of a computer

Description automatically generated with medium confidence

Once hadoop is installed create directory named LDZ and /LDZ/data in Hadoop and get the data in local system, unzip it and transfer it to hadoop .

### Download the zip file from the following location & move it to your Hadoop environment. Place this data under /LDZ/data/ in Hadoop

A screenshot of a computer

Description automatically generated with medium confidence

The zip file was downloaded into the ubuntu machine using **wget** command

### Transfer file to HDFS

File transferred to HDFS into the location /DWZ/data using put command in HDFS

Text

Description automatically generated

# Step 2:

## 3. Build a data warehouse (location:/DWZ) with data partitioned based on Creation Date and then Post Type.

### Install Hive in hadoo using following commands

1. Download 'apache-hive-1.2.1-bin' file from apache mirrors https://archive.apache.org/dist/hive/hive-1.2.1/, copy 'apache-hive-1.2.1-bin' file into this'/home/hadoop/work' directory and extract the tar files into same directory.

Using terminal:tar -xvzf apache-hive-1.2.1-bin

2. Open the '~/.bashrc' file using command: gedit ~/.bashrc in the terminal add below lines at the end of the document.

export HIVE\_HOME=/home/hadoop/work/apache-hive-1.2.1-bin

export PATH=$HIVE\_HOME/bin:$PATH

3. Copy mysql-connector-java-5.1.38 to /home/hadoop/work/apache-hive-1.2.1-bin/lib

Copy /home/hadoop/work/db-derby-10.11.1.1-bin/lib/derbyclient.jar to /home/hadoop/work/apache-hive-1.2.1-bin/lib

4. The following configuration files are required for hive to be run in different modes.

1.Hive-site.xml (Main configuration file)

2.hive-site.xml\_local (For this mode, copy paste this script into Main configuration file)

3.Hive-site.xml\_derby (For this mode, copy paste this script into Main configuration file)

4.hive-site.xml\_mysql (For this mode, copy paste this script into Main configuration file)

5. Open hive-site.xml from /home/hadoop/work/apache-hive-1.2.1-bin/conf. Edit configuration with below properties.

Once Hive is installed, add xml create database and table in Hadoop

A screenshot of a computer

Description automatically generated with medium confidence

We need serializer before creating table in hive, so download the serializer put it in hadoop and add jar file to hive providing the location

A screenshot of a computer

Description automatically generated with medium confidence

Create External table in hive

A screenshot of a computer

Description automatically generated with medium confidence

create external table posts( Id int, PostTypeId int, CreationDate string, Score int, ViewCount int, Body string,OwnerUserId int,OwnerDisplayName string, LastEditorUserId int,LastEditorDisplayName string, LastEditDate string, LastActivityDate string, Title string, Tags string, AnswerCount int, CommentCount int, FavoriteCount int, ContentLicense string )

ROW FORMAT SERDE 'com.ibm.spss.hive.serde2.xml.XmlSerDe'

WITH SERDEPROPERTIES (

"column.xpath.Id"="/row/@Id",

"column.xpath.PostTypeId"="/row/@PostTypeId",

"column.xpath.CreationDate"="/row/@CreationDate",

"column.xpath.Score"="/row/@Score",

"column.xpath.ViewCount"="/row/@ViewCount",

"column.xpath.Body"="/row/@Body",

"column.xpath.OwnerUserId"="/row/@OwnerUserId",

"column.xpath.OwnerDisplayName"="/row/@OwnerDisplayName",

"column.xpath.LastEditorUserId"="/row/@LastEditorUserId",

"column.xpath.LastEditorDisplayName"="/row/@LastEditorDisplayName",

"column.xpath.LastEditDate"="/row/@LastEditDate",

"column.xpath.LastActivityDate"="/row/@LastActivityDate",

"column.xpath.Title"="/row/@Title",

"column.xpath.Tags"="/row/@Tags",

"column.xpath.AnswerCount"="/row/@AnswerCount",

"column.xpath.CommentCount"="/row/@CommentCount",

"column.xpath.FavoriteCount"="/row/@FavoriteCount",

"column.xpath.ContentLicense"="/row/@ContentLicense"

)

STORED AS

INPUTFORMAT 'com.ibm.spss.hive.serde2.xml.XmlInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'

LOCATION '/DWZ/data'

TBLPROPERTIES (

"xmlinput.start"="

<row",

"xmlinput.end"="/>"

);

Load data into posts table

A screenshot of a computer

Description automatically generated with medium confidence

Show few data in hive in table posts

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

# Step 3:

## Querying Database:

What are the top 10 most answered questions in Stack Overflow posts for a particular creation date?

Query:

Select \* from posts where

creationdate ='2008-07-31T21:42:52.667'

ORDER BY answercount limit 10;

;

Text

Description automatically generated

What’s the percentage of Stack Overflow questions that went unanswered in say 2015?

Query:

Select \* from posts where creationdate like '20015 %' AND answercount=0;

# Step 4

## Automation

This process can be automated using pipeline. A batch of data can be uploaded daily and required transformation can be performed. After performing the transformation, the data will be loaded into the warehouse. From data stored in the warehouse’s analytics can be performed.

Diagram

Description automatically generated

# Conclusion

Hence the required architecture for the analysis of the stackoverflow data was built using

* Hadoop
* Hive

On Hive, mysql mode was used and the data was loaded in the external table and queries were performed to answer the given questions.