

Department of Computing

Academic Year 2022/2023

Infrastructure for Big Data – Assessment 3

**STUDENT NAME: M.S.Shruthi**

**STUDENT NUMBER: C00290801**

**COURSE NAME: Master of Science in Data Science**

**DEPARTMENT: Department of Computing and Data Science**

**SUBMITTED TO: Michael Gleeson**

**DATE OF SUBMISSION: December 23rd 2022.**

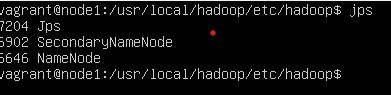
**Introduction:**

In this Project, I have built Big Data Infrastructure using Apache Hadoop which provides a software framework for distributed storage consisting of HDFS(Distributed File System), MapReduce(large-scale data processing), and Hadoop YARN(Managing computing resources). I will be providing a use case which typically involves small representative data set of the file by the name “Diabetes.csv” obtained from the Kaggle dataset in HIVE and demonstrated map-reduce Job.

Initially, the infrastructure has been set up manually, and tested with a small set of data with Map reduce jobs, and Hive Dataware housing SQL . Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

Further, used Vagrant to automate the setup of 3 Node VM’s , Installation of Hadoop, and its configuration and installation of HIVE. Vagrant uses “Provisioners” as building blocks to manage the development environments. Machine and software requirements are written in a file called "Vagrantfile" to execute the necessary steps in order to create a development-ready box. "Box" is a [format](https://en.wikipedia.org/wiki/File_format) and an [extension](https://en.wikipedia.org/wiki/Software_extension) (.box) for Vagrant environments that is copied to another machine in order to replicate the same environment.

**A**. Through Vagrant I have set up 3 VM hadoop clusters. Commands used are : Vagrant up, vagrant provision, vagrant halt, vagrant destroy. Here is jps status of namenode through vagrant.

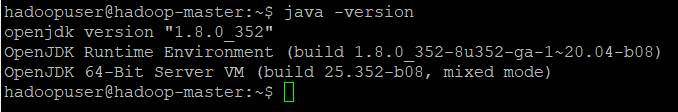


1. **Manual setup and Installation Steps:**

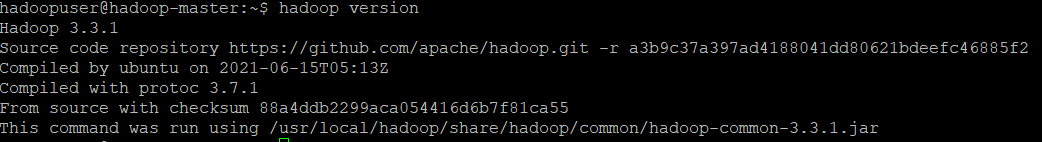
Ubuntu is installed on a 3 virtual machine nodes. To install Hadoop java needs to be installed.

The command to install java8 is: **sudo apt-get install openjdk-8-jdk**.

1. To verify version of the java installed: **java -version**.



1. Wget <https://mirrors.sonic.net/apache/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz>
2. To verify Hadoop version: hadoop version



1. To unzip Hadoop-3.3.1.tar.gz: tar xzf hadoop-3.3.1.tar.gz.
2. To rename the folder: sudo mv hadoop-3.3.1 hadoop
3. Enter java home variable in hadoop-env.sh.

Export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/ in ~/hadoop/etc/hadoop/hadoop-env.sh.



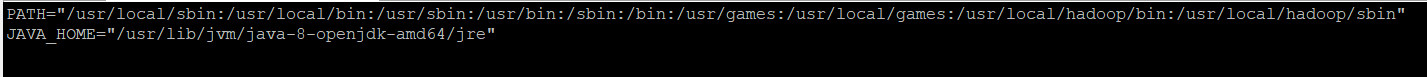
1. Change the **hadoop** folder directory to **/usr/local/Hadoop:**

sudo mv hadoop /usr/local/hadoop

1. Adding below configuration in sudo vi /etc/environment.

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"JAVA\_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"

1. Source / etc/environment



1. Adding user hadoopuser and execute below commands for hadoopuser

sudo adduser hadoopuser

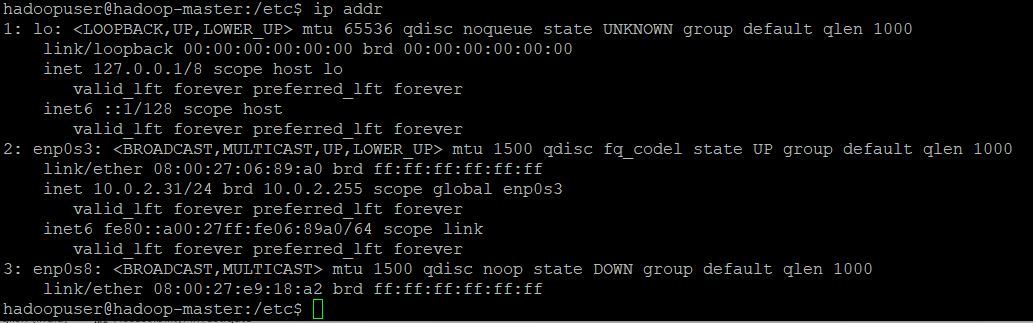
sudo usermod -aG hadoopuser hadoopuser

sudo chown hadoopuser:root -R /usr/local/hadoop/

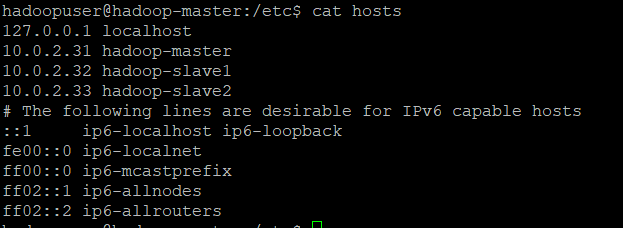
sudo chmod g+rwx -R /usr/local/hadoop/

sudo adduser hadoopuser sudo

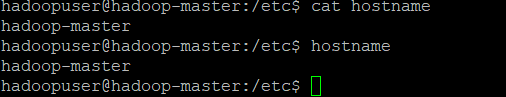
1. verify the ip addr.



1. Hosts file should look like this ,cd /etc/hosts:

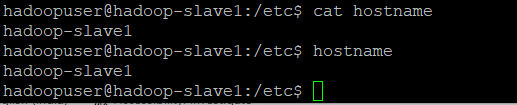


1. Clone master to 2 remaining slaves.
2. Cat hostname.

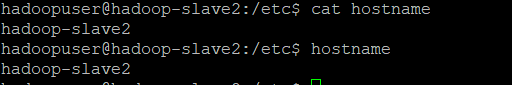


1. Reboot all of them so this configuration takes into effect.

Hadoop-slave1:



Hadoop-slave2:



1. Configured the SSH on hadoop-master, with the hadoopuser.

su – hadoopuser

generate ssh key: ssh-keygen -t rsa

Copied SSH key to all the users by this command:

ssh-copy-id hadoopuser@hadoop-master

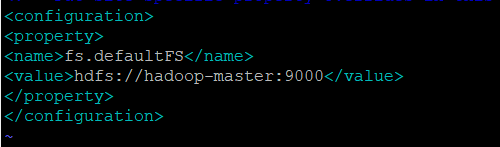
ssh-copy-id hadoopuser@hadoop-slave1

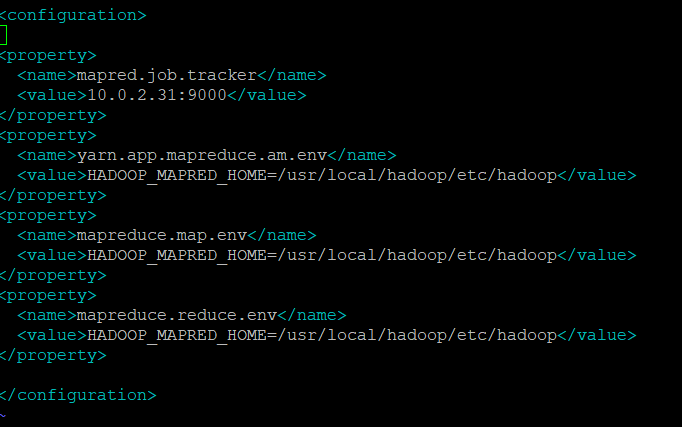
ssh-copy-id hadoopuser@hadoop-slave2

**2. HADOOP CONFIGURATION FILES**

In HDFS Name node is the master. Slaves are data nodes. The following files needs to be configure in master node to run the Hadoop cluster. XML stands for an extensible markup language. Everything encoded by tags.

1. **Core-site.xml** : Location: /etc/Hadoop/cor e-site.xml. The Master IP address/hostname and port is mentioned in core-site.xml.



1. **Mapred-site.xml**: Job tracker is mapred-site.xml. Adding properties inside the configuration tag. To specify the master Hostname (IP Address) and port that the MapReduce job tracker runs at. The mapred-site. xml file contains the configuration settings for MapReduce daemons; the job tracker and the task-trackers.
2. **HDFS-site.xml** - Configuring 3 properties. Location:/etc/hadoop/hdfs-site.xml.

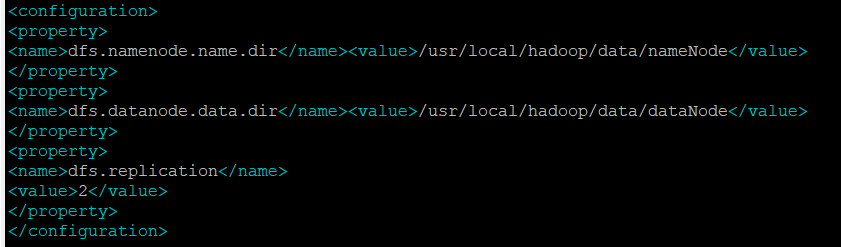
To configure replication factor.-->dfs.replication: distributed file system. Namenode is master node. In order to configure location of metdata .

Name of configuration is dfs.namenode.name.dir.

Complete file path: /usr/local/hadoop/data/namenode.

To configure data Node data storage location.

Dfs.datanode.name.dir : /usr/local/hadoop/data/datanode.



d. Masters - configure secondary name node which is running.

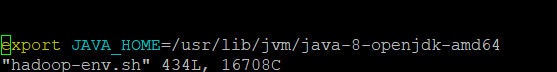
e. Slaves - contains list of all data nodes and addresses and task trackers.



f. hadoop-env.sh - Location: /etc/hadoop/hadoop-env.sh

Specify the java environment variable. JAVA\_HOME= /usr/lib/jvm/java-8-openjdk-amd64.

Hadoop runs on java.

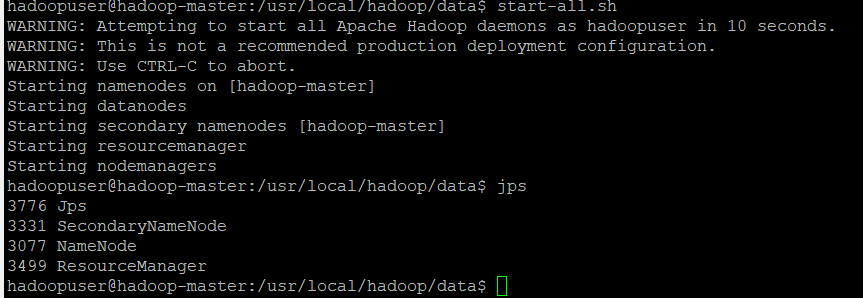


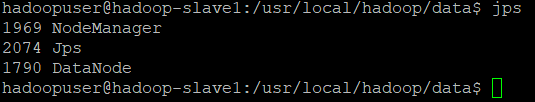
1. **Services Start**

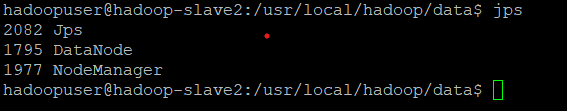
After completion of configuring xml files, Need to format namenode. Command to Format file system: **hadoop namenode -format**. Services has to be started.

**Commands: To start Hadoop cluster** – **start-all.sh**

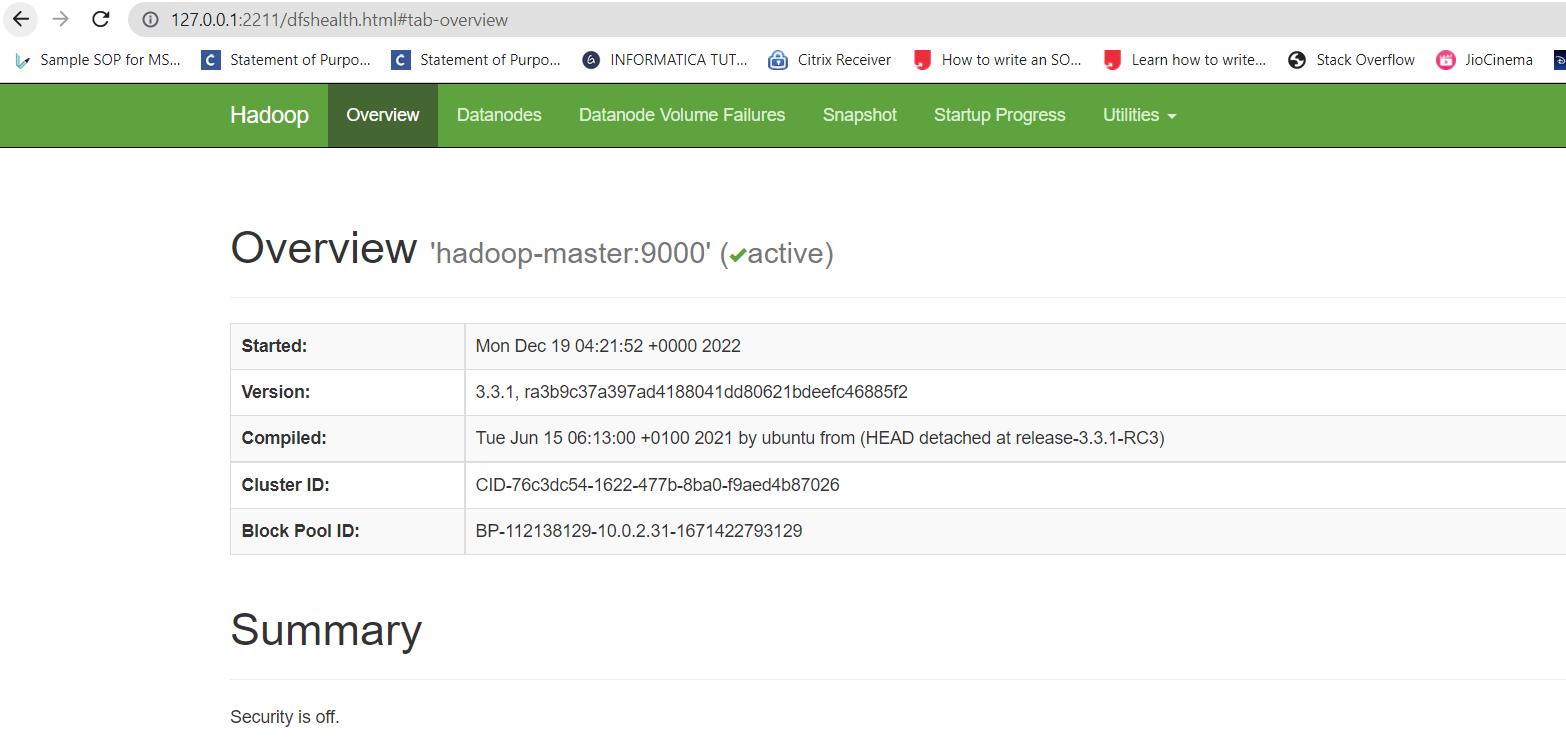
This will startup Namenode, secondary name node, data node,Job tracker, and Task tracker in a cluster. Java process status:jps. Will give information on services running with process id in a cluster.

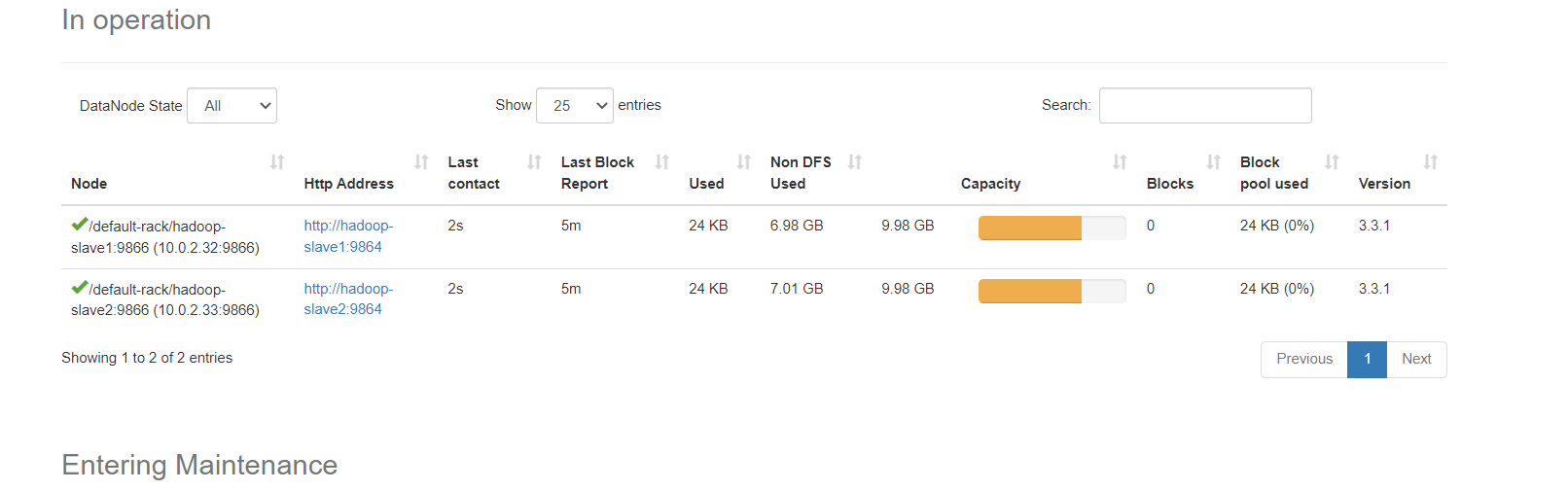


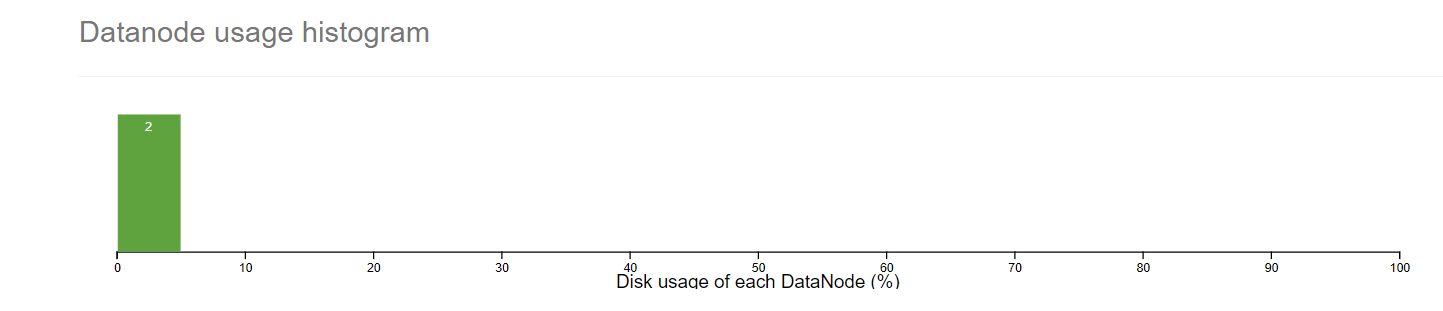




Web Browser :







1. **We need to copy the Hadoop Master configurations to the slaves**

scp /usr/local/Hadoop/etc/Hadoop/\* Hadoop-slave1:/usr/local/Hadoop/etc/Hadoop

scp /usr/local/Hadoop/etc/Hadoop/\* Hadoop-slave2:/usr/local/Hadoop/etc/Hadoop

source /etc/environment

1. **Configuring YARN:**

Export HADOOP\_HOME=”/usr/local/Hadoop”

Export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

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

Export HADOOP\_YARN\_HOME=$HADOOP\_HOME

**6. stop-all.sh**—all services will be stopped.

**7.Map-reduce job:**

HDFS - Hadoop Distributed File System. LFS – Local File System

Map-reduce job is basically Mapping of different computer, Data Processing using Map-reduce job. In this project, I have created sample file by name employeemaster.txt in Master Node and employeeslave.txt in slave data nodes and performed a map-reduce job for testing.

Uploaded file to HDFS , appended both files and executed map-reduce job on it.

Steps followed to perform Map-Reduce Job:

1. Create a sample File by the name **employeemaster.txt** in the master node locally.
2. Create a sample File by the name **employeeslave.txt** in the first data slave node.
3. Create a sample File by the name **employeeslave2.txt** in the second data slave node.
4. Created data directory in HDFS.
5. Copy from LFS master node to HDFS.(copyfromlocal)
6. Appends data files from slave node LFS file to HDFS.(appendToFile).
7. Execute map-reduce Job.

**Commands:**

hadoop fs -mkdir /data

vi employeemaster.txt

Hadoop fs -copyFromLocal employeemaster.txt /data

Similarly, sudo vi employeeslave.txt

hadoop fs -appendToFile employeeslave.txt /data/employeemaster.txt

sudo vi employeeslave2.txt

hadoop fs -appendToFile employeeslave2.txt /data/employeemaster.txt

Perform map-reduce job by executing following command:

Path: /usr/local/hadoop/share/hadoop/mapreduce

hadoop jar hadoop-mapreduce-examples-3.3.1.jar wordcount /data/employeemaster.txt /data/out\_employee

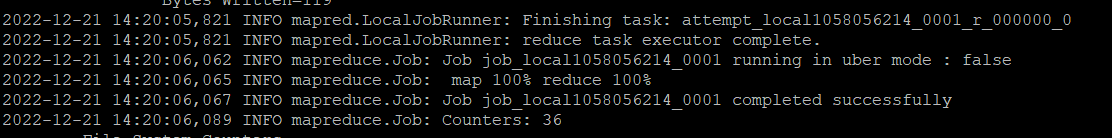
|  |  |  |
| --- | --- | --- |
| employeemaster.txt: | Slave1:Node file.employeeslave.txt | Slave2:Node file.employeeslave2.txt |
|  |  |  |

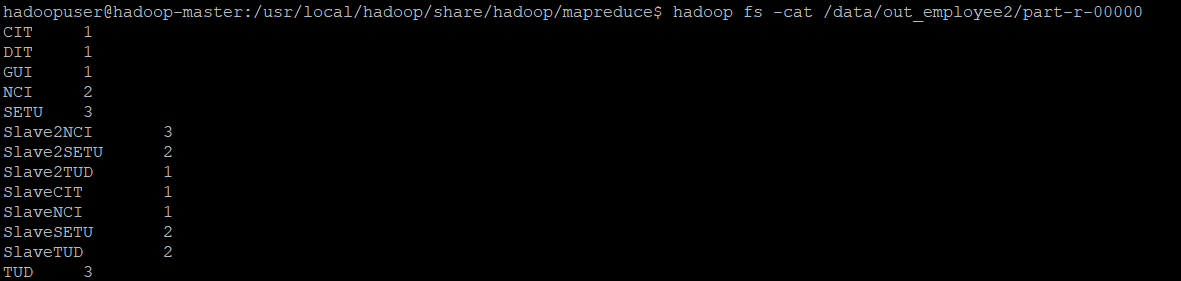
Map-reduce is written java language. Here , we are taking jar file and wordcount class and input file.

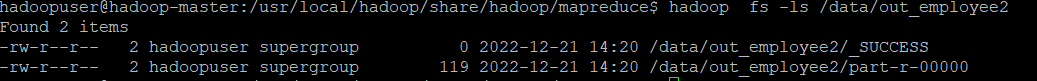
**hadoop jar hadoop-mapreduce-examples-3.3.1.jar wordcount /data/employeemaster.txt /data/out\_employee2**

**out\_employee2** directory has two files generated after map-reduce. One is SUCCESS Status. Other file has count of words .









**APACHE HIVE**

Apache Hive is a data warehousing tool built on top of Apache Hadoop. Hive provides an SQL -like interface and an abstracted query language (HiveQL) to query the data stored in databases and file systems that are integrated with Hadoop. Hive supports analysis of large datasets stored in the HDFS with compatibility for file systems such as Amazon S3.

**Steps followed to install HIVE and configuration**.

Hive installation: URL: <https://hive.apache.org/downloads.html> https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz

Extract tar file:tar -zxvf tar file name

sudo mv apache-hive-3.1.1-bin /usr/lib/hive

ls /usr/lib/hive

**SET up environment variable:**

vi ~/.bashrc

#SET HIVE\_HOME

Export HIVE\_HOME=”/usr/lib/hive/apache-hive-3.1.2.bin”

PATH=$PATH:$HIVE\_HOME/bin

Export PATH

Reload environment variable: Source ~/.bashrc

ls $HIVE\_HOME

edit hive configuration file.

Vi /usr/lib/hive/apache-hive-3.1.2-bin/conf/hive-env.sh

mv hive-env.sh.template hive-env.sh

ADD THE Following line which is home path for hdoop.

#Write directory path where Hadoop is installed

Export HADOOP\_HOME=/usr/local/hadoop

**Create Hive database directory on HDFS**

Hadoop fs -mkdir /usr/

Hadoop fs -mkdir /usr/hive

Hadoop fs -mkdir /usr/hive/warehouse

Hadoop fs -mkdir /tmp

Set READ/WRITE permission for warehouse and temporary directory

Hadoop fs -chmod g+w /usr/hive/warehouse

Hadoop fs -chmod g+w /tmp

**Edit hive environment variables**

Cd /usr/lib/hive/apache-hive-3.1.2-bin/conf

Cp hive-env.sh.template hive-env.sh

Vi /usr/lib/hive/apache-hive-3.1.2-bin/conf/hive-env.sh

**ADDED Below lines at the end of hive-env.sh file**

Export HADOOP\_HOME=/usr/local/Hadoop

Export HADOOP\_HEAPSIZE=512

Export HIVE\_CONF\_DIR=/usr/lib/hive/apache-hive-3.1.2-bin

Export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Export YARN\_HOME=$HADOOP\_YARN\_HOME

Set hive-env.sh as executable : Chmod +x hive-env.sh

**To enable logging when starting hive ,I have renamed as below**

Mv hive-log4j2.properties.template hive-log4j2.properties

Mv hive-exec-log4j2.properties.template hive-exec-log4j2.properties

**Change log level to ERROR only**

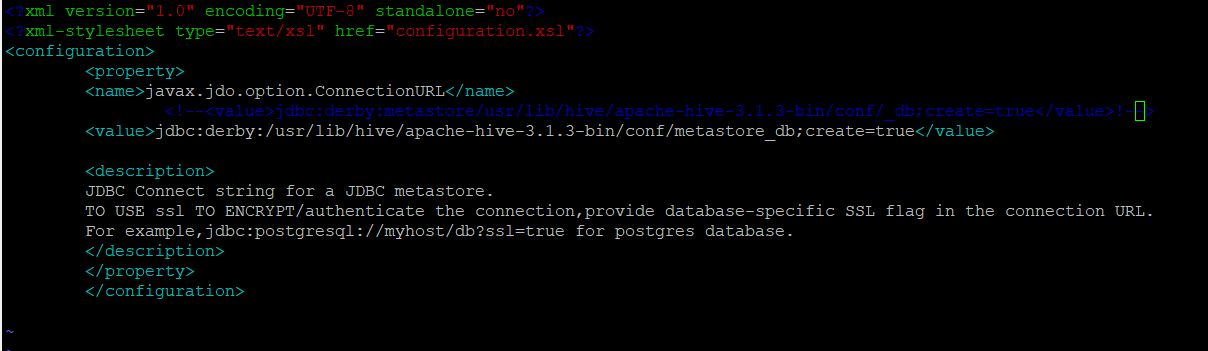
Vi hive-log4j2.propertis

Change status =INFO line to status =ERROR

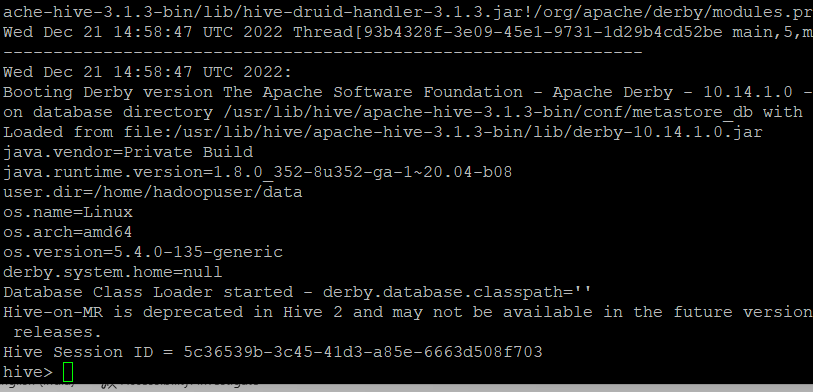
Vi hive-exec-log4j2.properties

Change status =INFO line to status =ERROR

Create new hive-site.xml, vi $HIVE\_HOME/conf/hive-site.xml to add metadata store db config.

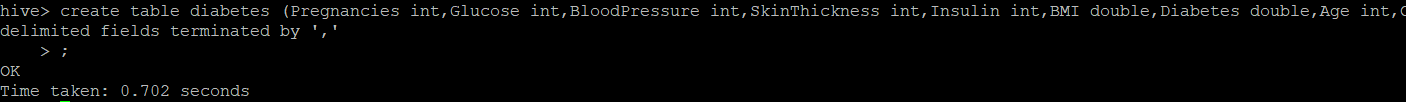


Hive console: Log into hive console .



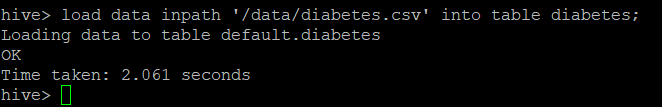
**Create table by name diabetes corresponding to the columns in “diabetes.csv”.**

Create Query: **create table diabetes (Pregnancies int,Glucose int,BloodPressure int,SkinThickness int,Insulin int,BMI double,Diabetes double,Age int,Outcome int)row format delimited fields terminated by ','**

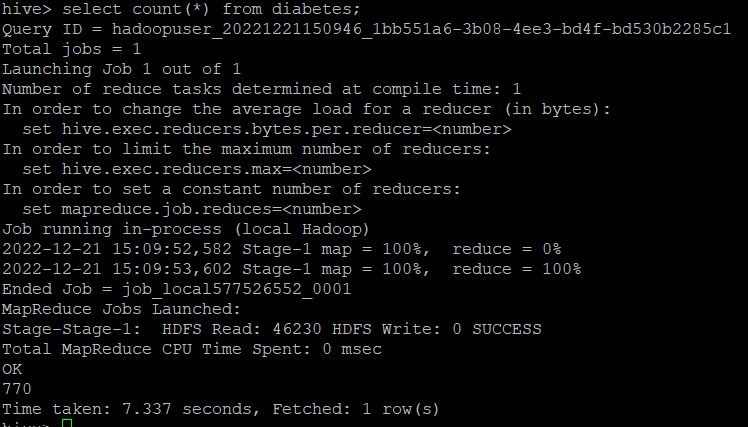


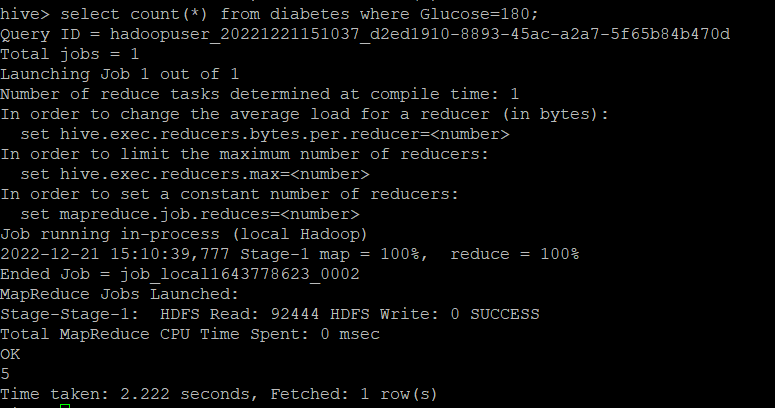
Load data into table diabetes.

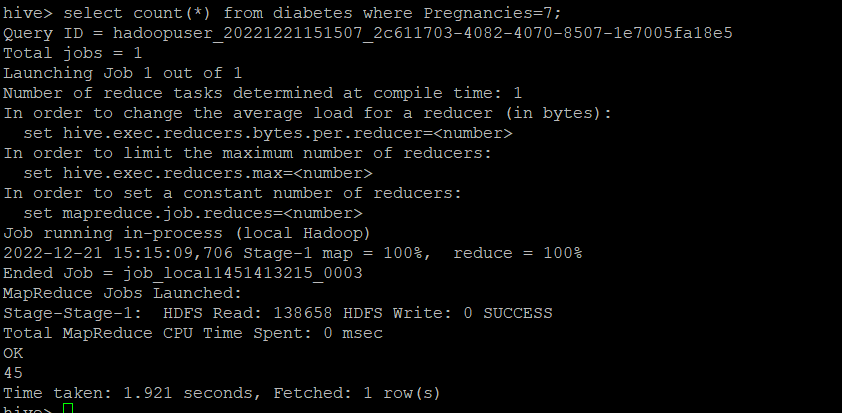
**load data inpath '/data/diabetes.csv' into table diabetes;**



**Performed data Retrieval operations on various criteria .**







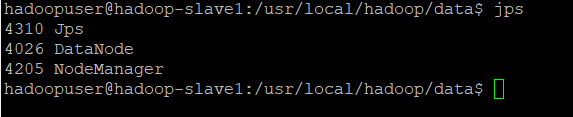
**Issues faced:**

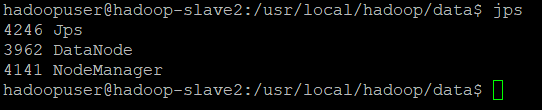
1. While HDFS System is configured and started services.datanode was not displaying.This is due to permission issue. Data Node is supposed to be Up and running in slave nodes.

Solution: Deleted old files in dataNode directory and created new directory .

“/usr/local/hadoop/data/dataNode”,created new directory dataNode. Stopped all services.stop-all.sh Executed namenode format : hdfs namenode -format.

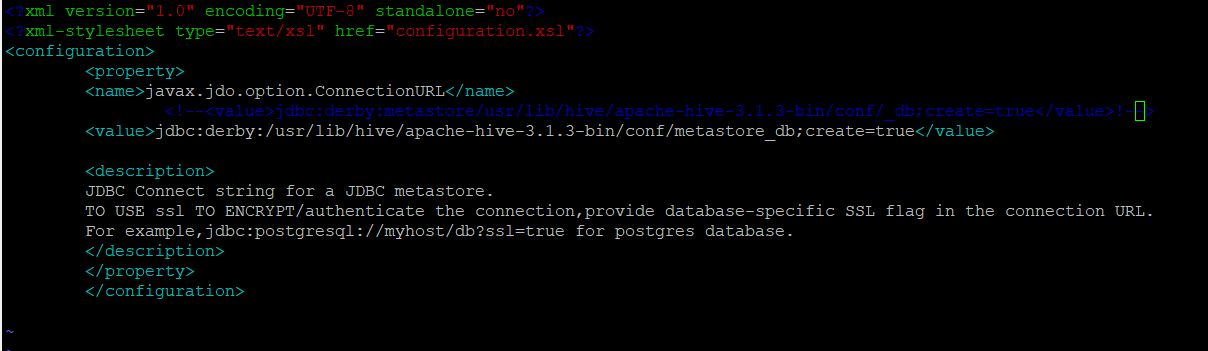
Started all services again.: start-all.sh It is resolved.





1. Hive console was throwing Semantic Exception.Console was not displaying.

Solution: Configured metadata store db .Deleted old metadata store db .



**References:**

https://ishan.co/installing-hadoop/

https://github.com/ishansharma/vagrant-hadoop

<https://github.com/njvijay/vagrant-jilla-hadoop>

<https://medium.com/@jootorres_11979/how-to-set-up-a-hadoop-3-2-1-multi-node-cluster-on-ubuntu-18-04-2-nodes-567ca44a3b12>

<https://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/>

<https://nesusws-tutorials-bd-dl.readthedocs.io/en/latest/hands-on/hadoop/install/>

https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.0.0.2/bk\_installing\_manually\_book/content/rpm-chap6-4.html

https://www.youtube.com/results?search\_query=data+ingress+csv+file+hive++on+multi+node+hadoop

<https://www.youtube.com/watch?v=A2JoxsWwvZ4>

<https://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/>

https://github.com/luckyfengyong/vagrant-hadoop

https://ishan.co/installing-hadoop/

<https://www.youtube.com/watch?v=-dX0tkXiVMY>

https://www.youtube.com/watch?v=hzwSk1saR\_s

https://www.youtube.com/watch?v=x4JQYyKSXhc

https://www.youtube.com/watch?v=Tb3V7jtBeB8&list=PL3GJWzvY8ndF3DMB4KrPr6E3TTSBPZpGR

https://dzone.com/articles/install-and-configuration-of-apache-hive-312-on-mu

<https://gautambangalore.medium.com/install-and-configuration-of-apache-hive-on-multi-node-hadoop-cluster-534a28372fc8>

<https://www.youtube.com/watch?v=HqRP6O-M5aY>

https://www.ibm.com/docs/en/spectrum-scale-bda?topic=STXKQY\_BDA\_SHR/bl1bda\_mapredyarnclientconfig.htm