

ITA-6008 Big Data Analytics Assignment - 1

By :

Rajat Singh 22MCA0139

Submitted to :

**Prof. POUNAMBAL M**

1. **Procedure to install the Hadoop in your system.**

# Prerequisite to Hadoop Installation

* 1. You have installed Ubuntu 22 Desktop version in your Virtual Machine
  2. You have installed Java (jdk 8) in your Ubuntu system.
  3. Check your hostname is Ubuntu

$ hostname --should output Ubuntu

# Linux Configuration Before Hadoop Installation

We will setup single node Hadoop cluster using a dedicated Hadoop user.

1. Login as Root
2. Adding a dedicated user called hduser

2. Create a Group called Hadoop

sudo addgroup Hadoop

1. Create an User hduser

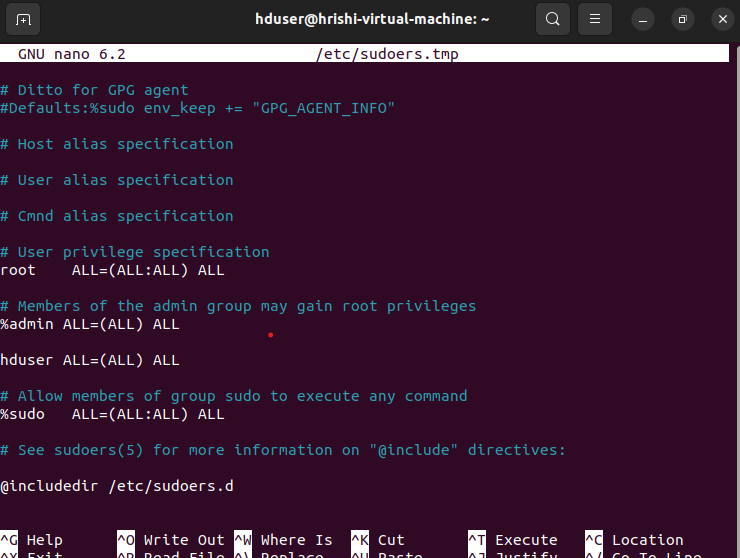
sudo adduser hduser

1. Add houser to hadoop group

sudo adduser houser Hadoop

1. Add the 'hduser' to sudoers list so that hduser can do admin tasks. sudo visudo

houser ALL=(ALL) ALL



1. Logout Your System and login again as hduser.
2. Configuring SSH

sudo apt-get install openssh-server

1. Generate SSH for communication

hduser@ubuntu:~$ ssh-keygen

1. Copy Public Key to Authorized key file & edit the permission

hduser@ubuntu:~Scat~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

ho&ser@ubuntu:~$chmod 700 ~/.ssh/auth 11.Start SSH

If ssh is not running, then run it by giving the below command

hduser@ubuntu:~$ sudo /etc/init.d/ssh restart Enter your Password(hadoop)

1. Test Your SSH Connectivity

hduser@ubuntu:~$ ssh localhost

1. Test Your SSH Connectivity hduser@ubuntu:-S ssh localhost

# Download Hadoop

1. Download hadoop-3.3.1.tar.gz and save it to hduser/Desktop. https://downloads.apache.org/hadoop/
2. move the above downloaded file to /us/local/ Open Terminal(Ctr|+Alt+T)

$ sudo mv ~/Desktop/hadoop-3.3.1.tar.gz /us/local/ cd /usr/local

sudo tar -xvf hadoop-3.3.1.tar.gz sudo rm hadoop-3.3.1.tar.gz sudo ln -s hadoop-3.3.1 hadoop

sudo chown -R hduser:hadoop hadoop-3.3.1 sudo chmod 777 hadoop-3.3.1

1. Edit hadoop-env.sh and configure Java.

$ sudo vim /ust/local/hadoop/etc/hadoop/hadoop-env.sh

export HADOOP\_OPTS=Djava.net.preferiPv4Stack=true export HADOOP\_HOME\_WARN\_SUPPRESS-"TRUE"

export JAVA\_HOME=/us/local/java/jdk

3. Update $HOME/.bashrc

# Set Hadoop-related environment variables export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_MAPRED\_HOME=${HADOOP\_HOME} export HADOOP\_COMMON\_HOME=${HADOOP\_HOME} export HADOOP\_HDFS\_HOME=${HADOOP\_HOME} export HADOOP\_YARN\_HOME=${HADOOP\_HOME}

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

# Native Path export

HADOOP\_COMMON\_LIB\_NATIVE\_DIR=${HADOOP\_PREFIX}/lib/native export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_PREFIX/lib"

# Set JAVA\_HOME (we will also configure JAVA\_HOME directly for Hadoop later on)

export JAVA\_HOME=/usr/local/java/jdk

# Some convenient aliases and functions for running Hadoop-related commands unaliasfs&> /dev/null

aliasfs="hadoop fs" unaliashls&> /dev/null aliashls="fs -ls"

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

OME/sbin

1. Update yarn-site.xml

$sudo vim /us/local/hadoop/etc/hadoop/yarn-site.xmI

<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>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

<description>Whether virtual memory limits will be enforced for containers</description>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>4</value>

<description>Ratio between virtual memory to physical memory when setting memory limits for containers</description>

</property>

1. Update core-site.xml file

$ sudo vim /ust/local/hadoop/etc/hadoop/core-site.xmI

<property>

<name>hadoop.tmp.dir</name>

<value>/app/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

<description>default host and port</description>

</property>

<property>

<name>hadoop.proxyuser.hduser.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.hduser.groups</name>

<value>\*</value>

</property>

1. Create the above temp folder and give appropriate permission

sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop -R /app/hadog sudo chmod 750 /app/hadoop/tmp

1. Edit mapred-site.xml

sudo vim /us/local/hadoop/etc/hadoop/mapred-site.xml

<property>

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

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>localhost:10020</value>

<description>Host and port for Job History Server (default

0.0.0.0:10020)</description>

</property>

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

<property>

<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

1. Create a temporary directory which will be used as base location for

sudo mkdir -p /us/local/hadoop tmp/hdfs/namenode sudo,mkdir -p /ust/local/hadoop\_tmp/hdfs/datanode sudo chown hduser:hadoop -R /us/local/hadoop tmp/

1. Update hdfs-site.xmI file

$ sudo vim /us/local/hadoop/etc/hadoop/hdfs-site.xml

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/usr/local/hadoop\_tmp/hdfs/namenode</value>

</property>

<property>

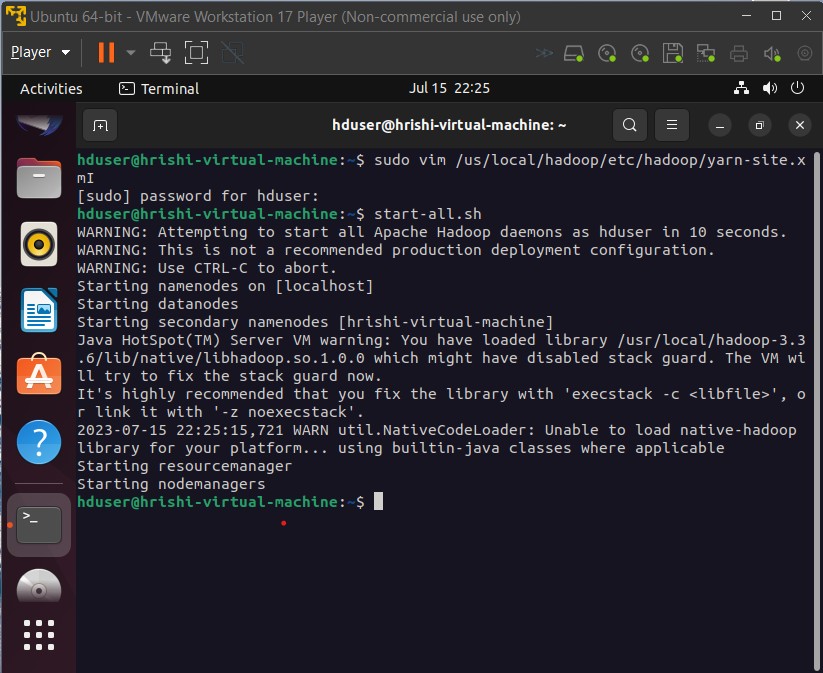
<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop\_tmp/hdfs/datanode</value>

</property> 11.Format your namenode

$ hadoop namenode -format 12.Starting your single-node cluster

$ start-all.sh



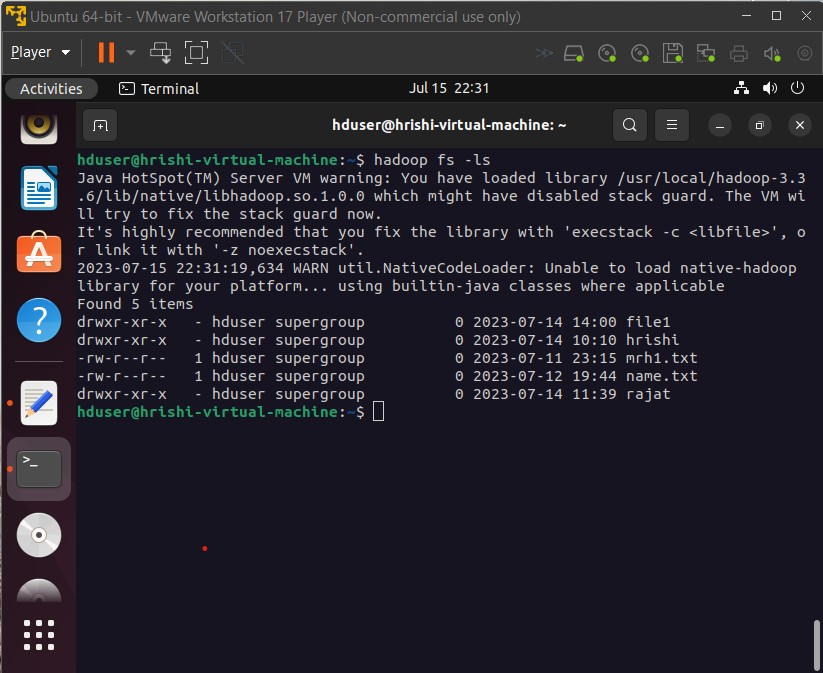
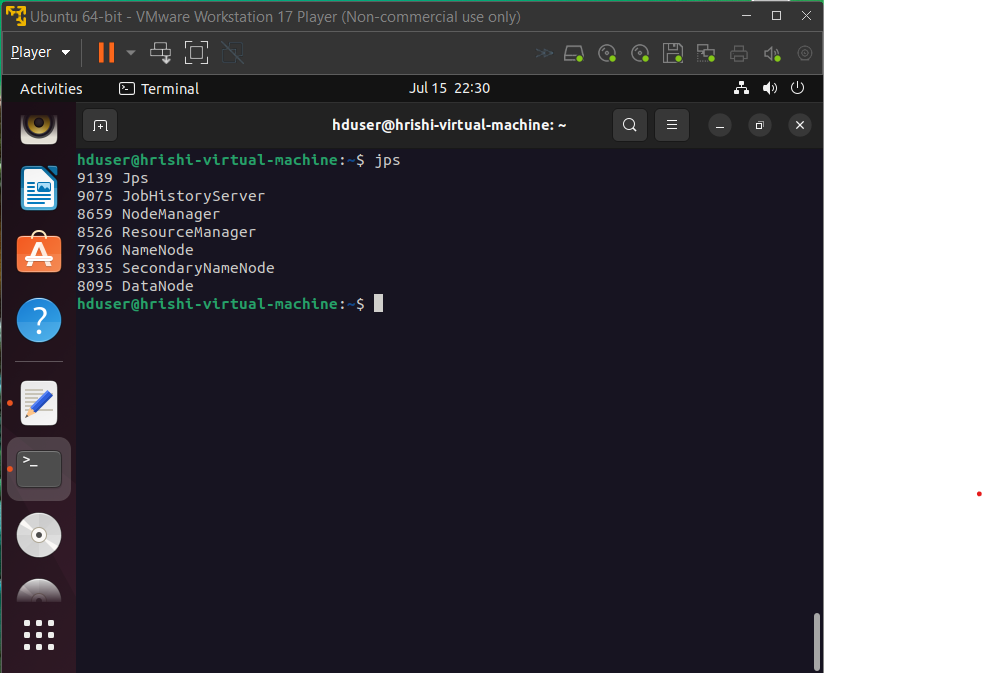
1. Start your history-server.

$ mr-jobhistory-daemon.sh start historyserver

$ mr-jobhistory-daemon.sh stop historyserver

1. Check if all the necessary hadoop daemon is running or not

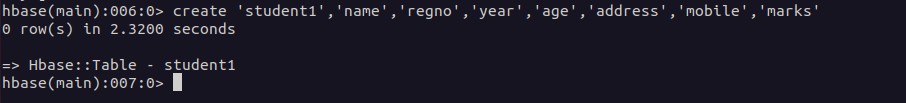
$ jps



# 2. Creation of sample table in HABASE with 7 attributes with 2 sub columns and displaying of valid data.

Table Creation :

create 'student1', 'name', 'regno', 'year", 'age', 'address', 'mobile', 'marks'



put ‘student1’, 'row1', 'Name', 'John'

put ' student1', 'row1', 'RegNo', '2021001' put 'student1', 'row1', 'Year', '2023'

put ‘student1’, 'row1', 'Age', '25'

put ‘student1’, 'row1', 'Mobile', '9876543210'

put ‘student1’, 'row1', 'Address', '123 Main Street' put ‘student1’, 'row1', 'Marks:Marks1', '80'

put ‘student1’, 'row1', 'Marks:Marks2', '75'

