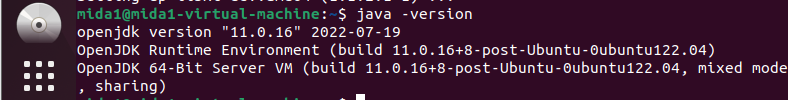
**Lab 1: Hadoop Installation and Hive Installation**

Section 1: Hadoop installation

1. Install the java (highlighted line) and check the revision installed.

Graphical user interface, text

Description automatically generated



1. Install Hadoop 2.10.1 or 3.3.4 (2.9.1 is no longer available)
2. In this example, I fetch Hadoop 2.10.1, extract the file and move it into “/usr/local”, then downloaded latest Hadoop version 3.3.4 to try the performance of latest version. Here is the screenshot for old version.

Text

Description automatically generated

After typing in

tar -xzvf hadoop-2.10.1 tar.gz, then type in following command,



1. Configure Hadoop’s Java Home



Enter into the file hadoop-env.sh,



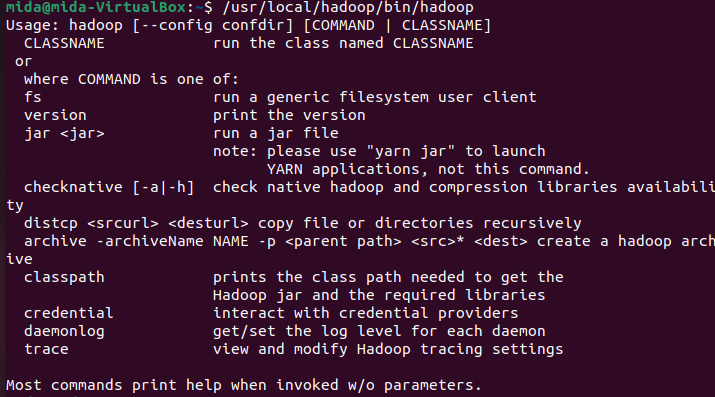
then change the

export JAVA\_HOME into following statement:



save the file then exit the hadoop-env.sh

1. Run Hadoop:



1. Run example MapReduce program to make sure Hadoop is functioning properly:
2. Create a directory called input in home directory. Since I created previously, therefore, it indicates file exists.



1. Copy Hadoop’s configuration files into it to use those files as our data

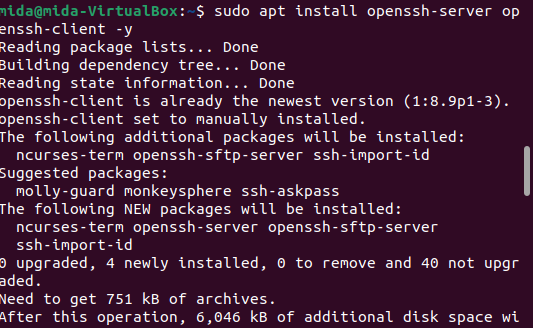


1. Download Hadoop example
2. Configure SSH

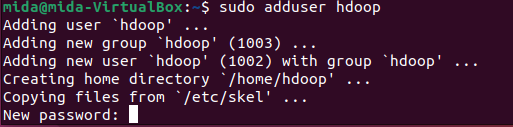
In order to manage nodes in a cluster, Hadoop requires SSH access.

First, we need to add a user. This time, we add a user called hduser

1. Install SSH:



1. To add user in the Hadoop environment



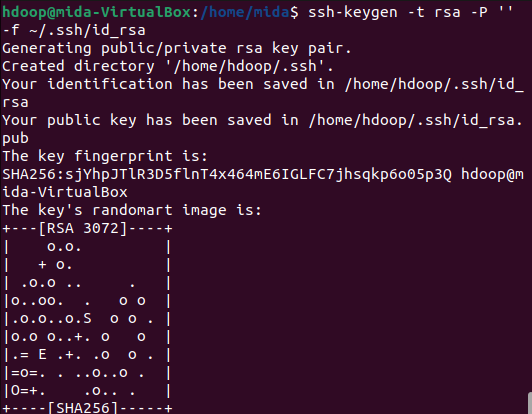
1. To provide sudo permission to add user



1. To verify the current login, type in “whoami”, the answer should be the new user created: hdoop



1. To configure passwordless SSH for Hdoop user

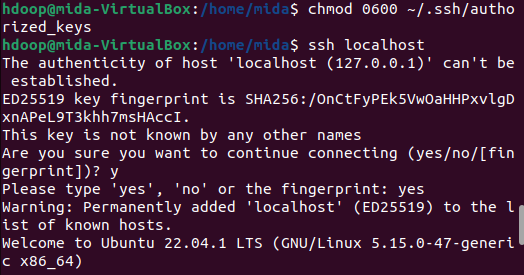


1. To copy the public keys from id\_rsa.pub to authorized-keys



1. To change the permissions to hdoop user, run the chmod command:

Then verify that the localhost setup is done properly.



1. Download hadoop under user hdoop:





1. Setup Hadoop in standalong/single node, we need to modify the six environment configuration files. They are .bashrc, hadoop-env.sh, core-site.xml, hdfs-site.xml, mapred-site-xml and yarn-site.xml
2. .bashrc



Type following codes in the end of the .bashrc file:

#Hadoop Related Options

export HADOOP\_HOME=/home/hdoop/hadoop-3.2.1

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

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

export HADOOP\_OPTS=”-Djava.library.path=$HADOOP\_HOME/lib/native”

To apply changes to the current hadoop environment run the following command:

$ source ~/.bashrc

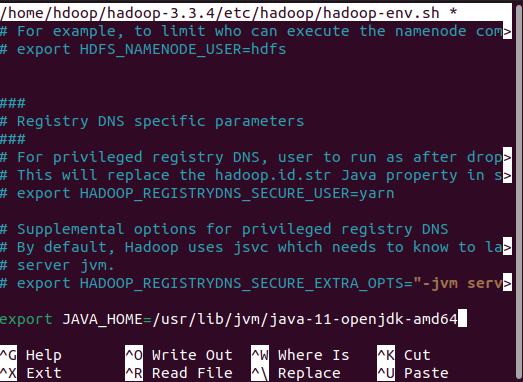
1. To make configuration changes in hadoop-env.sh file

Text

Description automatically generated

Now, enter following command and add java path at the end of the file.





1. Setup configuration changes in core-site.xml file, we need to add default path for the temporary directory and add HDFS URL.



Enter the following code at the end of the file, between <configuration> and <.configuration> tag, save and exit.

<property>

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

<value>/home/hdoop/tmpdata</value>

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

</property>

<property>

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

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

<description>The name of the default file system></description>

</property>

1. Change configuration in hdfs-site.xml file



Add the following code in the end of the file. Between tag <configuration> and </configuration>

<property>

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

<value>/home/hdoop/dfsdata/namenode</value>

</property>

<property>

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

<value>/home/hdoop/dfsdata/datanode</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

#Above 1 indicates number of replication factor that is 1.

1. Make changes in mapred-site-xml file



Add following code at the end of the mapred-site-xml file:

Text

Description automatically generated

1. Modify the yarn file



Add following code between tag <configuration> and </configuration>

<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.resourcemanager.hostname</name>

<value>127.0.0.1</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

</property>

1. Now we finished the Hadoop configuration, we need to format NameNode first time. To format NameNode, run the following command:

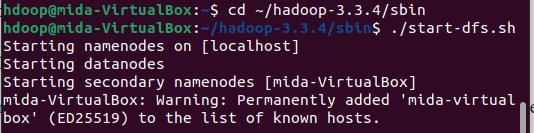
Text

Description automatically generated

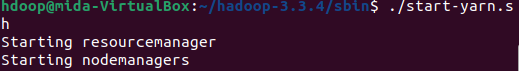
1. To start Hadoop cluster traverse to the ~/hadoop-3.3.4/sbin/directory and execute the following commands.



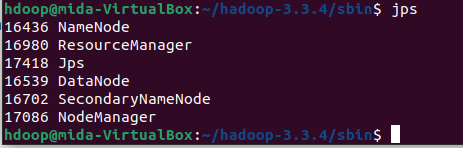
Output:



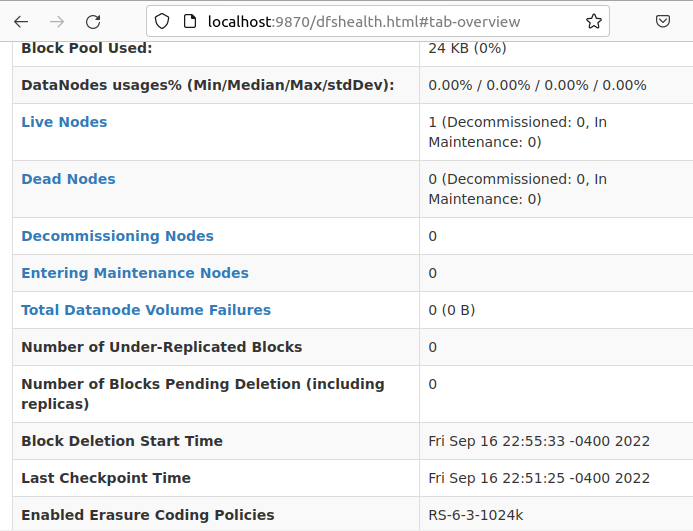
Then start the YARN resource and NodeManagers:



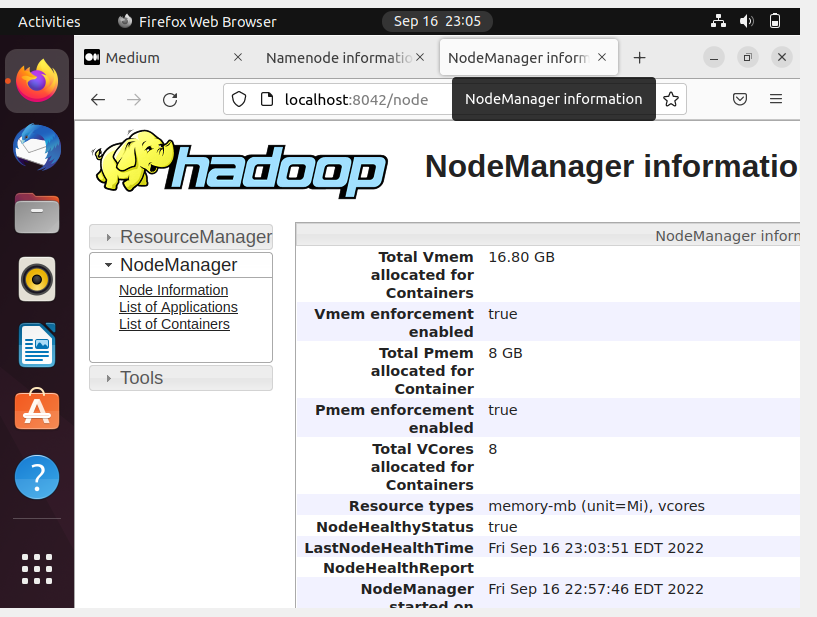
To check and verify all the Hadoop daemons are started, we can use the command $jps to verify



1. We can also review the result from browser at default port 9870:

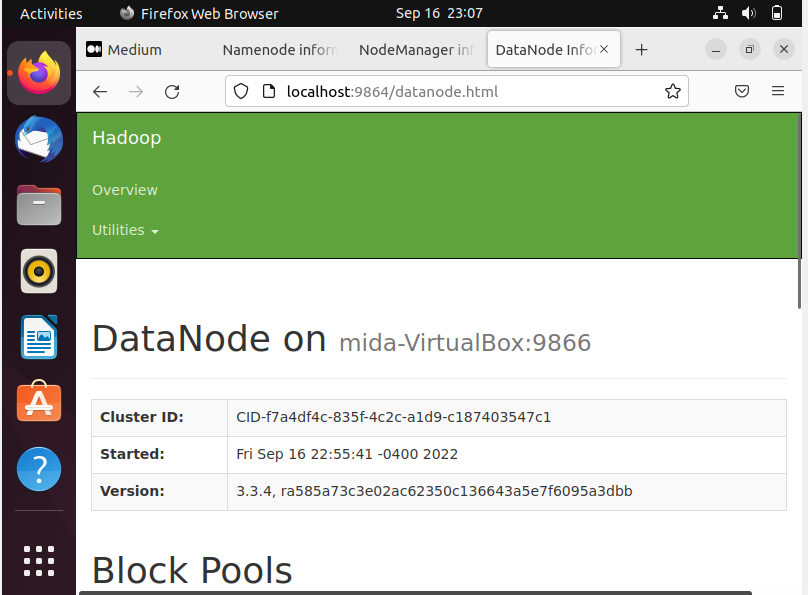


We can get information about the cluster and all applications by access to port 8042. Run <http://localhost:8042/> in the browser



To get details of Hadoop node, we can also access port 9864

The output is:



1. To stop services of Hadoop cluster, execute the command ./stop-all.sh

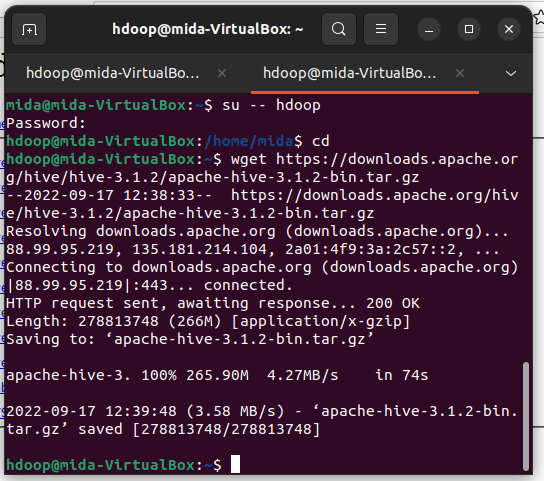
Text

Description automatically generated

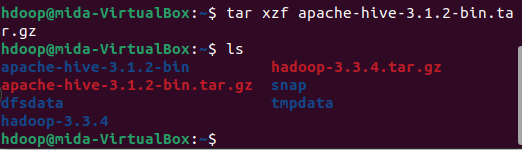
Section 2: HIVE installation

HIVE 2.3.3 is an old revision. This time, we will install latest version HIVE 3.1.2 instead.

1. Go to website <https://download.apache.org/hive/> to find the latest version of the hive.
2. Download HIVE 3.1.2, then unzip the file under user hdoop.

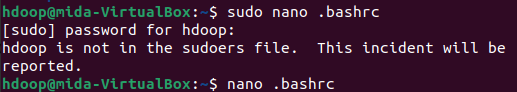


Using tar command to unzip the file, then check the installed list with command ls.



1. Edit Environment settings.
2. Edit .bashrc file

Enter into .bashrc file, if sudo nano doesn’t work, because it is not the root system, can use nano directly.



Add 2 export statement under original code,

Text

Description automatically generated

Save then exit the file, run the source ~/.bashrc to execute the file.



1. Edit hive\_config.sh file



Add the HADOOP\_HOME variable and the full path to the Hadoop directory:



1. Create Hive Directories in HDFS

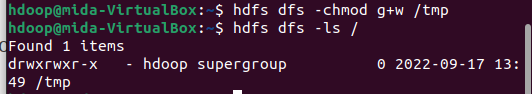
Create two separate directories to store data in the HDFS layer:

* The temporary, tmp directory is going to store the intermediate results of Hive processes,
* The warehouse directory is going to store the Hive related table.

1. Create tmp Directory

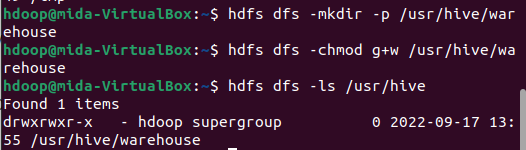


Add write and execute permissions to tmp group members, then use ls command to check whether we made it correctly:



1. Create warehouse Directory

Similar as adding tmp directory, we use the same commands to add warehouse directory.

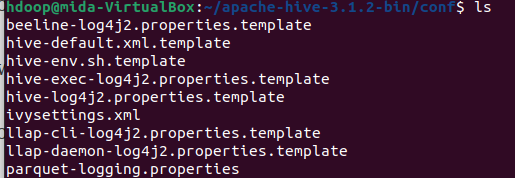


1. Configure hive-site.xml file

Use the command to locate the correct file:



List the files contained in the folder using ls command, and find out the file hive-default.xml.template



Use the hive-default.xml.template to create the hive-site.xml file

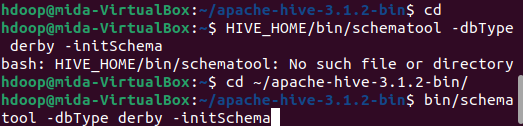


Access the hive-site.xml file:

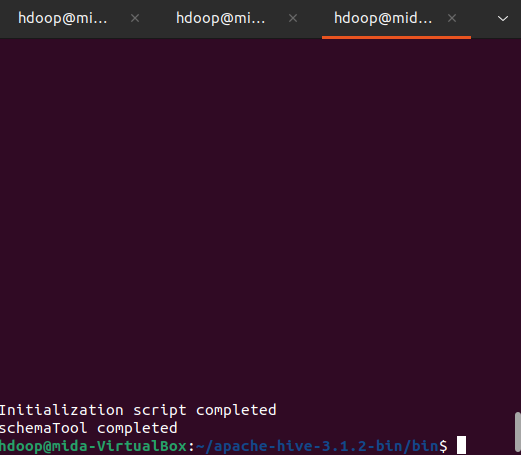


1. Initiate Derby Database

Apache Hive uses the Derby database to store metadata. Initiate the Derby database, from the Hive bin directory using the schematool command:



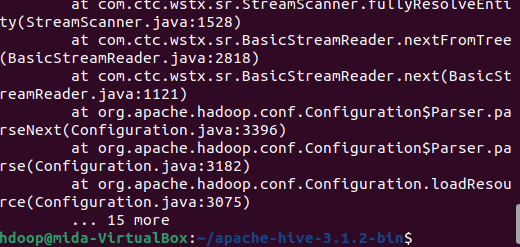
Output: wait until showing schemaTool completed



1. Launching Hive



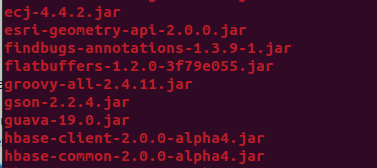
Output:



However, error message showing out. We locate the guava jar file in the Hive lib directory:



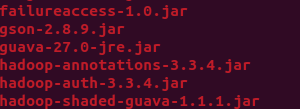
Output for the guava version is 19.0:



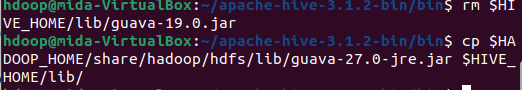
Then we check the guava jar file in the hdfs lib as well,



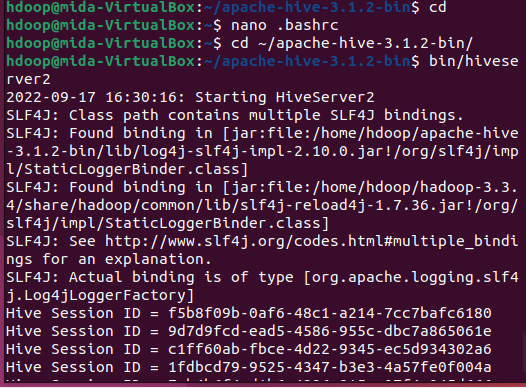
Output for the guava version is 27.0



The two listed versions are not compatible, that’s why it doesn’t work properly. We remove the existing guava file from the Hive lib directory, then copy the guava file from the Hadoop lib directory to the Hive lib directory.



Now we do the test again, and the output is: with another screen working on the link practice, the hiveserver2 is listen to it and record it.



Then open a new terminal window, put following commands in the bin folder, and we can use hive successfully. (if doesn’t work, please stop-all.sh, then start-all.sh, to start a new namenode to let it work properly)

