-- Install single node hadoop cluster

Pre-requisites:

1. Ubuntu OS image desktop 18.04, 20.04 or 22.04
2. VMware Workstation or VMware player /virtualbox
3. (virtualization) from the CPU should be enabled.

// Install jdk

sudo apt update

sudo apt install openjdk-11-jdk

// check java version

java -version

// install openssl client and server

sudo apt-get install openssh-client openssh-server

// Create hadoop user

sudo adduser hadoop

// add new password

// configure SSH key-based authentication

su - hadoop

// add user to sudoers privilege

sudo su

adduser [username] sudo

//generate public and private key pair

ssh-keygen -t rsa

// append the generated public keys from id\_rsa.pub to authorized\_keys

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

chmod 0600 ~/.ssh/authorized\_keys

// verify passwordless SSH authentication

ssh localhost

// Installing Hadoop

su - hadoop

// download the latest version of hadoop using the wget command

https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz

// once downloaded, extract the downloaded file

tar -xvzf hadoop-3.3.4.tar.gz

// rename the extracted directory to hadoop

mv hadoop-3.3.4 hadoop

// configure the hadoop and java environment variables

vim ~/.bashrc

/\*

Hadoop excels when deployed in a fully distributed mode on a large cluster of networked servers. However, if you are new to Hadoop and want to explore basic commands or test applications, you can configure Hadoop on a single node.

This setup, also called pseudo-distributed mode, allows each Hadoop daemon to run as a single Java process. A Hadoop environment is configured by editing a set of configuration files:

bashrc

hadoop-env.sh

core-site.xml

hdfs-site.xml

mapred-site-xml

yarn-site.xml

\*/

// configure hadoop environment

sudo vim .bashrc

// define the hadoop environment

#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"

// apply the changes

source ~/.bashrc

// check the default java path of your system

readlink -f /usr/bin/java | sed "s:bin/java::"

// edit hadoop-env.sh

sudo vim $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

// uncomment the $JAVA\_HOME variables

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

//find the OpenJDK directory

readlink -f /usr/bin/javac

// edit core-site.xml file

sudo vim $HADOOP\_HOME/etc/hadoop/core-site.xml

// replace HDFS URL

<configuration>

<property>

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

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

</property>

<property>

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

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

// edit hdfs-site.xml

sudo vim $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

// add the following configurations to the file

<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>

</configuration>

// edit mapred-site.xml file

sudo vim $HADOOP\_HOME/etc/hadoop/mapred-site.xml

// add the following configurations

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

// edit yarm-site.xml file

sudo vim $HADOOP\_HOME/etc/hadoop/yarn-site.xml

<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>

</configuration>

// format HDFS namenode

hdfs namenode -format

// Start hadoop cluster

start-dfs.sh

// start the YARN resources

start-yarn.sh

// check all the resources are running

jps

// access hadoop UI from browser

http://localhost:9870 - namenode

http://localhost:9864 - datanode

http://localhost:8088 - YARN resource manager

// Make HDFS directories required to execute Mapreduce jobs

bin/hdfs dfs -mkdir /user

bin/hdfs dfs -mkdir /user/<username>

// Copy the input files into the distributed filesystem

bin/hdfs dfs -mkdir input

bin/hdfs dfs -put etc/hadoop/\*.xml input

// Run some examples

bin/hadoop jar $HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar grep input output 'dfs[a-z.]+'

// examine the output files

bin/hdfs dfs -get output output

cat output/\*

//or view the output files on the distributed filesystem

hdfs dfs -cat output/\*

//stop the dfs daemons when done

stop-dfs.sh