**EXPERIMENT 1**

| Objective :  Theory:  Performance:  **Deliverables:**  **Summary:** | To **install the Hadoop** ecosystem and **execute** basic file operation commands.  Hadoop is a free, open-source, and Java-based software framework used for the storage and processing of large datasets on clusters of machines. It uses HDFS to store its data and process this data using MapReduce. It is an ecosystem of Big Data tools that are primarily used for data mining and machine learning.  Modules of Hadoop  **HDFS**: **H**adoop **D**istributed **F**ile **S**ystem. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.  **Yarn**: **Y**et **a**nother **R**esource **N**egotiator is used for job scheduling and managing the cluster.  **Map Reduce**: This is a framework which helps Java programs to do the parallel computation on data using key value pairs. The Map task takes input data and converts it into a data set which can be computed in Key value pairs. The output of Map task is consumed by reduce task and then the output of reducer gives the desired result.  **Hadoop Common**: These Java libraries are used to start Hadoop and are used by other Hadoop modules.  **Step 1 – Installing Java**  Hadoop is written in Java and supports only Java version 8. Hadoop version 3.3 and the latest also support Java 11 runtime as well as Java 8.   | $ sudo apt update  $ sudo apt install openjdk-11-jdk | | --- |   Once installed, verify the installed version of Java with the following command:   | $ java -version | | --- |   You should get the following output:    **Step 2 – Create a Hadoop User**  It is a good idea to create a separate user to run Hadoop for security reasons. Run the following command to create a new user with name hadoop\_yourname:   | $ sudo adduser hadoop\_yourname | | --- |   Provide and confirm the new password as shown below:  Though you can provide any password of your choice, it is recommended that you keep the password as 123456 as it is easy to remember.    **Step 3 – Configure SSH Key-based Authentication**  Next we need to configure passwordless SSH authentication for the local system.  First, change the user to hadoop\_yourname with the following command:   | $ su - hadoop\_yourname | | --- |   install the OpenSSH server and client:   | $ sudo apt install openssh-server openssh-client -y | | --- |   Generate Public and Private Key Pairs   | $ ssh-keygen -t rsa | | --- |   You will be asked to enter the filename. Just press Enter to complete the process:  You will see similar output  cat  Append the generated public keys from id\_rsa.pub to authorized keys and set proper permission:   | $ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  $ chmod 640 ~/.ssh/authorized\_keys | | --- |   Next, verify the passwordless SSH authentication with the following command:   | $ ssh localhost | | --- |   You will be asked to authenticate hosts by adding RSA keys to known hosts. Type yes and hit Enter to authenticate the localhost:    **Step 4 – Installing Hadoop**  Once you’ve installed Java, you can download Apache Hadoop and all its related components, including Hive, Pig, Sqoop, etc. You can find the latest version on the official Hadoop’s download page. Make sure to download the binary archive (not the source).  First, change the user to hadoop\_yourname with the following command:   | $ su - hadoop\_yourname | | --- |   download Hadoop 3.3.6   | $ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz | | --- |   unzip it to a folder on your hard drive   | $ tar xzf hadoop-3.3.6.tar.gz | | --- |   Rename the extracted folder to remove version information. This is an optional step, but if you don’t want to rename, then adjust the remaining configuration paths.   | $ mv hadoop-3.3.6 hadoop | | --- |   Next, you will need to configure Hadoop and Java Environment Variables on your system. Open the ~/.bashrc file in your favorite text editor   | $ nano ~/.bashrc | | --- |   Once the editor opens append lines given below in the editor   | export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64  export HADOOP\_HOME=/home/hadoop\_yourname/hadoop  export HADOOP\_INSTALL=$HADOOP\_HOME  export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_HOME=$HADOOP\_HOME  export HADOOP\_HDFS\_HOME=$HADOOP\_HOME  export HADOOP\_YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native  export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin  export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native" | | --- |   To save and exit the editor **ctrl** + **X** + **Y**  Load the above configuration in the current environment   | $ source ~/.bashrc | | --- |   Configure JAVA\_HOME in hadoop-env.sh file. Edit the Hadoop environment variable file in the text editor:   | $ nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh | | --- |   Search for the “export JAVA\_HOME” and configure it with given value   | $ export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64 | | --- |   To save and exit the editor **ctrl** + **X** + **Y** Step 5 – Configuring Hadoop Create the namenode and datanode directories inside Hadoop home directory:   | $ mkdir -p ~/hadoopdata/hdfs/namenode  $ mkdir -p ~/hadoopdata/hdfs/datanode | | --- |   Edit the core-site.xml file and update with your system hostname:   | $ nano $HADOOP\_HOME/etc/hadoop/core-site.xml | | --- |   Add following code to the core-site,xml   | <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://0.0.0.0:9000</value>  </property>  </configuration> | | --- |   To save and exit **ctrl + X + Y**  **Open hdfs-site.xml and add given code**   | **$ nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml**  <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.name.dir</name>  <value>file:///home/hadoop\_yourname/hadoopdata/hdfs/namenode</value>  </property>  <property>  <name>dfs.data.dir</name>  <value>file:///home/hadoop\_yourname/hadoopdata/hdfs/datanode</value>  </property>  </configuration> | | --- |   To save and exit **ctrl + X + Y**  Edit the mapred-site.xml file:   | $ nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml  <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property><property>  <name>mapreduce.application.classpath</name>  <value>$HADOOP\_MAPRED\_HOME/share/hadoop/  mapreduce/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/lib/\*</value>  </property>  </configuration> | | --- |   To save and exit **ctrl + X + Y**  Edit the yarn-site.xml file   | $ nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml  <configuration>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <property> #copy this part only  <name>yarn.nodemanager.env-whitelist</name>  <value>JAVA\_HOME,HADOOP\_COMMON\_HOME,  HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,  CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,  HADOOP\_HOME,PATH,LANG,TZ,HADOOP\_MAPRED\_HOME</value>  </property>  </configuration> | | --- |   **Step 6 – Start Hadoop Cluster**  Before starting the Hadoop cluster. You will need to format the Namenode as a hadoop user.   | $ hdfs namenode -format | | --- |   You should get following output    After formatting the Namenode, run the following command to start the Hadoop cluster   | $ hdfs namenode -format | | --- |   To start the Hadoop services   | $ start-all.sh | | --- |   To see if all the required processes are up and running, run following command   | $ jps | | --- |   You should see all the running processes    S**tep 7 – Access Hadoop Namenode and Resource Manager** - http://localhost:9870  And the Hadoop application page is available at <http://localhost:8088>  With this we have successfully installed Hadoop on Ubuntu  Now let’s try some file operation commands.   | $ hdfs dfs -mkdir /test1  $ hdfs dfs -mkdir /logs | | --- |    run the following command to list the above directory:   | $ hdfs dfs -ls / | | --- |   To copy a file to Hadoop HDFS file system from your local machine use following command   | $ hdfs dfs -put /home/file1.txt /test1  This command will copy the file present at given location on the local machine to test1 folder created on Hadoop file system. | | --- |  1. Step wise Screen shots of the Hadoop Installation 2. Screen shots of the Hadoop hdfs commands execution     Write summary in your own words. |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**To be inserted into mapred.xml**

<configuration>

<property>

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

<value>yarn</value>

</property><property>

<name>mapreduce.application.classpath</name>

<value>$HADOOP\_MAPRED\_HOME/share/hadoop/

mapreduce/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/lib/\*</value>

</property>

</configuration>

**For yarn-site.xml**

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property> #copy this part only

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

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_HOME,PATH,LANG,TZ,HADOOP\_MAPRED\_HOME</value>

</property>

</configuration>