**DEPARTMENT OF INFORMATION TECHNOLOGY**

**ExperimentNo.2**

|  |  |
| --- | --- |
| Semester | B.E. Semester VIII–Information Technology |
| Subject | BigData Analytics |
| Subject Professor In-charge | Prof. Deepali Vora |
| Assisting Teachers | Prof. Deepali Vora |
| Laboratory | Lab 011B |

|  |  |  |
| --- | --- | --- |
| Student Name | Ritupurna Meha Shahi | |
| Roll Number | 12101A0043 | |
| Grade and Subject Teacher’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Experiment Number | 2 | |
| Experiment Title | Hadoop installation | |
| Resources / Apparatus Required | Hardware: | Software: |
| Objectives  (Skill Set / Knowledge Tested / Imparted) | To study and install Hadoop | |
| Theory of operation | **HADOOP:**  Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.  **COMPONENTS OF HADOOP:**  **HDFS** (storage) and **MapReduce** (processing) are the two core components of Apache Hadoop. The most important aspect of Hadoop is that both HDFS and MapReduce are designed with each other in mind and each are co-deployed such that there is a single cluster and thus pro­vides the ability to move computation to the data not the other way around. Thus, the storage system is not physically separate from a processing system.  **Hadoop Distributed File System (HDFS)**  HDFS is a distributed file system that provides high-throughput access to data. It provides a limited interface for managing the file system to allow it to scale and provide high throughput. HDFS creates multiple replicas of each data block and distributes them on computers throughout a cluster to enable reliable and rapid access.   * NameNode is the master of the system. It maintains the name system (directories and files) and manages the blocks which are present on the DataNodes. * DataNodes are the slaves which are deployed on each machine and provide the actual stor­age. They are responsible for serving read and write requests for the clients. * Secondary NameNode is responsible for performing periodic checkpoints. In the event of NameNode failure, you can restart the NameNode using the checkpoint.   **MapReduce**  MapReduce is a framework for performing distributed data processing using the MapReduce programming paradigm. In the MapReduce paradigm, each job has a user-defined map phase (which is a parallel, share-nothing processing of input; followed by a user-defined reduce phase where the output of the map phase is aggregated). Typically, HDFS is the storage system for both input and output of the MapReduce jobs.  The main components of MapReduce are as described below:   * JobTracker is the master of the system which manages the jobs and resources in the clus­ter (TaskTrackers). The JobTracker tries to schedule each map as close to the actual data being processed i.e. on the TaskTracker which is running on the same Data Node as the underlying block. * TaskTrackers are the slaves which are deployed on each machine. They are responsible for running the map and reduce tasks as instructed by the JobTracker. * JobHistoryServer is a daemon that serves historical information about completed applications. Typically, JobHistory server can be co-deployed with Job­Tracker, but we recommend to run it as a separate daemon.   **STEPS FOR INSTALLATION:**  How to Setup Hadoop 2.6.0 (Single Node Cluster) on CentOS/RHEL and Ubuntu :  Step 1: Installing Java  # java -version  java version "1.8.0\_66"  Java(TM) SE Runtime Environment (build 1.8.0\_66-b17)  Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)  How to Install JAVA 8 (JDK 8u66) on CentOS/RHEL and Fedora  For 64Bit  # cd /opt/  # wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u66-b17/jdk-8u66-linux-x64.tar.gz"  # tar xzf jdk-8u66-linux-x64.tar.gz  For 32Bit  # cd /opt/  # wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u66-b17/jdk-8u66-linux-i586.tar.gz"  # tar xzf jdk-8u66-linux-i586.tar.gz  # java -version  Step 2: Creating Hadoop User  # adduserhadoop  # passwdhadoop  # su - hadoop  $ ssh-keygen -t rsa  $ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  $ chmod 0600 ~/.ssh/authorized\_keys  $ ssh localhost  $ exit  Step 3. Downloading Hadoop 2.6.0  $ cd ~  $ wget http://apache.claz.org/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz  $ tar xzf hadoop-2.6.0.tar.gz  $ mv hadoop-2.6.0 hadoop  Step 4. Configure Hadoop Pseudo-Distributed Mode  4.1. Setup Environment Variables  First we need to set environment variable uses by hadoop. Edit ~/.bashrc file and append following values at end of file.  export HADOOP\_HOME=/home/hadoop/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 YARN\_HOME=$HADOOP\_HOME  export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native  export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin  Now apply the changes in current running environment  $ source ~/.bashrc  Now edit $HADOOP\_HOME/etc/hadoop/hadoop-env.sh file and set JAVA\_HOME environment variable. Change the JAVA path as per install on your system.  export JAVA\_HOME=/opt/jdk1.8.0\_66/  4.2. Edit Configuration Files  Hadoop has many of configuration files, which need to configure as per requirements of your hadoop infrastructure. Lets start with the configuration with basic hadoop single node cluster setup. first navigate to below location  $ cd $HADOOP\_HOME/etc/hadoop  Edit core-site.xml  <configuration>  <property>  <name>fs.default.name</name>  <value>hdfs://localhost:9000</value>  </property>  </configuration>  Edit hdfs-site.xml  <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.name.dir</name>  <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>  </property>  <property>  <name>dfs.data.dir</name>  <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>  </property>  </configuration>  Edit mapred-site.xml  <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration>  Edit yarn-site.xml  <configuration>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  </configuration>  4.3. Format Namenode  $ hdfsnamenode -format  Sample output:  15/02/04 09:58:43 INFO namenode.NameNode: STARTUP\_MSG:  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  STARTUP\_MSG: Starting NameNode  STARTUP\_MSG: host = svr1.tecadmin.net/192.168.1.133  STARTUP\_MSG: args = [-format]  STARTUP\_MSG: version = 2.6.0  ...  ...  15/02/04 09:58:57 INFO common.Storage: Storage directory /home/hadoop/hadoopdata/hdfs/namenodehas been successfully formatted.  15/02/04 09:58:57 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid>= 0  15/02/04 09:58:57 INFO util.ExitUtil: Exiting with status 0  15/02/04 09:58:57 INFO namenode.NameNode: SHUTDOWN\_MSG:  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SHUTDOWN\_MSG: Shutting down NameNode at svr1.tecadmin.net/192.168.1.133  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Step 5. Start Hadoop Cluster  $ cd $HADOOP\_HOME/sbin/  $ start-dfs.sh  Now run start-yarn.sh script.  $ start-yarn.sh  Step 6. Access Hadoop Services in Browser  Hadoop NameNode started on port 50070 default. Access your server on port 50070 in your favorite web browser.  http://svr1.tecadmin.net:50070/  Now access port 8088 for getting the information about cluster and all applications  http://svr1.tecadmin.net:8088/  Access port 50090 for getting details about secondary namenode.  http://svr1.tecadmin.net:50090/  Access port 50075 to get details about DataNode  http://svr1.tecadmin.net:50075/  Step 7. Test Hadoop Single Node Setup  7.1 – Make the HDFS directories required using following commands.  $ bin/hdfsdfs -mkdir /user  $ bin/hdfsdfs -mkdir /user/hadoop  7.2 – Now copy all files from local file system /var/log/httpd to hadoop distributed file system using below command  $ bin/hdfsdfs -put /var/log/httpd logs  7.3 – Now browse hadoop distributed file system by opening below url in browser.  http://svr1.tecadmin.net:50070/explorer.html#/user/hadoop/logs  7.4 – Now copy logs directory for hadoop distributed file system to local file system.  $ bin/hdfsdfs -get logs /tmp/logs  $ ls -l /tmp/logs/ | |
| Output/  Screenshots | C:\Users\Ritu\Downloads\Screenshot.png | |
| Conclusion | Thus we have successfully studied installation of Hadoop. | |