|  |  |
| --- | --- |
| **Ex. No :** | **8** |
| **Date** | **01/06/25** |

|  |
| --- |
| **INSTALL AND CONFIGURE A HADOOP SINGLE-NODE CLUSTER AND EXECUTE A SAMPLE APPLICATION (WORDCOUNT) TO VALIDATE THE SETUP.** |

|  |  |
| --- | --- |
| **AIM:** | To install and configure a Hadoop single-node cluster and execute a sample application (WordCount) to validate the setup. |
| **SOFTWARE/TOOLS USED:** | * **Operating System:** Ubuntu 20.04 LTS (or compatible Linux distribution) * **Java Development Kit (JDK):** Java 8 (OpenJDK 1.8.0) * **Hadoop Version:** Apache Hadoop 3.x (stable release) * **Terminal/Command Line Interface** * **Text Editor:** gedit, nano, or vim * **SSH (Secure Shell):** Enabled for localhost communication |
| **APPLICATIONS :** |  **Big Data Processing:** Facilitates distributed processing of large datasets using the MapReduce programming model.   **Data Analysis and Mining:** Suitable for batch processing tasks such as word frequency analysis, log parsing, and ETL operations.   **Educational and Research Purpose:** Provides a simulated Hadoop environment for learning, testing, and prototyping in academic settings. |
| **PROCEDURE:** |  |

|  |
| --- |
| **Hadoop Installation Procedure** |
| **Step 1: Download Java 8 Package** Download the required Java 8 Development Kit (JDK) and save the archive file in the home directory.  (Ensure compatibility with Hadoop by using Java 1.8.x only) |
| **Step 2: Extract Java Tar File** Execute the following command to extract the Java archive:  bash  tar -xvf jdk-8u101-linux-i586.tar.gz  This unpacks the Java installation directory, which will later be configured in the environment path. |
| **Figure: Hadoop Installation – Extracting Java Files** |
| **Step 3: Download Hadoop 2.7.3 Package** Use the following command to download Hadoop 2.7.3 from the Apache archives:  bash  wget <https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz> |
| **Figure:** Hado op Installation – Downloading Hadoop |
| **Step 4: Extract Hadoop Tar File** Unpack the Hadoop archive using the following command:  bash  tar -xvf hadoop-2.7.3.tar.gz  This will extract the Hadoop directory (hadoop-2.7.3) in the current location, which will be used to configure and launch the Hadoop environment. |
| *Fig: Hadoop Installation – Extracting Hadoop Files* **Step** |
| **Step 5: Configure Environment Variables** Update the system’s environment variables to include the paths to the installed **Java** and **Hadoop** directories by modifying the .bashrc file. |
| Command to open .bashrc file: |
| vi ~/.bashrc |
|  |
| Add the following lines at the end of the .bashrc file. Update the paths according to the actual installation directories: |
| |  | | --- | | # Set JAVA environment  export JAVA\_HOME=~/jdk1.8.0\_101  export PATH=$JAVA\_HOME/bin:$PATH  # Set Hadoop environment  export HADOOP\_HOME=~/hadoop-2.7.3  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=$HADOOP\_HOME  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin | |
|  |
| Save the .bashrc file and close the editor. |
|  |
| **Step 6:** Apply Environment Variable Changes  To apply the changes made to the .bashrc file in the current terminal session, use the source command: |
| Command:  source ~/.bashrc |
|  |
| **Step 7: Verify Java and Hadoop Installation** After setting the environment variables, verify whether **Java** and **Hadoop** are correctly installed and accessible through the terminal.  **Check Java version:**  bash  java -version |
|  |
| *Fig: Hadoop Installation – Checking Java Version* |
| Check Hadoop version:  hadoop version |
|  |
|  |
| *Fig: Hadoop Installation – Checking Java Version* |
| **Step 6: Edit the Hadoop Configuration Files** Navigate to the directory containing all configuration files necessary for setting up the Hadoop environment. |
| Command:  cd ~/hadoop-2.7.3/etc/hadoop/ |
|  |
| To list all available configuration files within the Hadoop directory:  **Command:**  ls |
|  |
|  |
| **Figure:** Hadoop Installation – Hadoop Configuration Files Directory Listing |
| This directory contains key configuration files such as:   * core-site.xml * hdfs-site.xml * mapred-site.xml * yarn-site.xml * hadoop-env.sh   These configuration files must be edited to set up the Hadoop environment as a single-node cluster. |
| **Step 7: Configure** core-site.xml The core-site.xml file defines core Hadoop parameters and specifies the NameNode address within the cluster. It includes I/O settings shared by both HDFS and MapReduce frameworks.  **Location:**  ~/hadoop-2.7.3/etc/hadoop/core-site.xml |
|  |
|  |
|  |
| *Fig: Hadoop Installation – Configuring core-site.xml* |
| Command to Open: |
| vi core-site.xml |
| Within the <configuration> tag, insert the following property block to define the default file system:     |  | | --- | | <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost:9000</value>  </property>  </configuration> | |
| </property>  </property>  1  2  3  4  5  6  7  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>fs.default.name</name>  <value>hdfs://localhost:9000</value>  </property>  </configuration> |
| *Fig: Hadoop Installation – Configuring core-site.xml* |
|  fs.defaultFS specifies the Hadoop Distributed File System (HDFS) URL.   localhost refers to the single-node setup.   Port 9000 is the default listening port for the NameNode. |
|  |
| S**Step 8: Configure** hdfs-site.xml The hdfs-site.xml file defines configuration parameters for HDFS components such as the NameNode, DataNode, and Secondary NameNode. It includes settings for block replication, storage directories, and other filesystem properties. |
| **Location:**  bash  ~/hadoop-2.7.3/etc/hadoop/hdfs-site.xml |
| **Command to Open:**  bash  vi hdfs-site.xml |
|  |
|  |
|  |
|  |
| Within the <configuration> tag, insert the following property blocks to set the replication factor and define storage directories: |
| <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///home/hduser/hadoopdata/hdfs/namenode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///home/hduser/hadoopdata/hdfs/datanode</value>  </property>  </configuration> |
|  dfs.replication defines the number of data copies stored in HDFS. A value of 1 is optimal for a single-node cluster.   dfs.namenode.name.dir and dfs.datanode.data.dir specify local directories for storing metadata and block data, respectively. Ensure these directories exist or are created before formatting HDFS. |
| 1  2  3  4  5  6  7  8  9  10  11  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.permission</name>  <value>false</value>  </property>  </configuration> |
| *Fig: Hadoop Installation – Configuring hdfs-site.xml* |
| **Step 9: Configure** mapred-site.xml The mapred-site.xml file specifies the configuration for MapReduce applications, including execution mode, resource allocation, and job scheduling. This configuration is critical for defining how MapReduce jobs are executed within the Hadoop ecosystem.  **Note:** In Hadoop 2.x distributions, the mapred-site.xml file may not be available by default. Instead, a template file named mapred-site.xml.template is provided. It must be copied and renamed before editing. |
| **Commands:**  cd ~/hadoop-2.7.3/etc/hadoop/  cp mapred-site.xml.template mapred-site.xml  vi mapred-site.xml |
| **Add the following content inside the <configuration> tag:**  xml  CopyEdit |
| <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration> |
|  |
|  |
|  |
| **Fig: Hadoop Installation – Configuring yarn-site.xml** |
| 1  2  3  4  5  6  7  <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration> |
|  |
| **Step 11: Configure** hadoop-env.sh The hadoop-env.sh file contains essential environment variable declarations required for the execution of Hadoop services. One of the most critical variables is the JAVA\_HOME path, which must be set to the directory of the installed Java Development Kit (JDK).  **Purpose:** Defines the Java environment required by Hadoop daemons during runtime. |
| Command: vi hadoop-env.sh |
| Edit or append the following line: export JAVA\_HOME=/home/username/jdk1.8.0\_101 |
| Replace /home/username/jdk1.8.0\_101 with the actual path where Java is extracted.  **Save and exit** the file after modification.  **Figure:** Editing hadoop-env.sh to set JAVA\_HOME  This step ensures that all Hadoop components reference the correct Java environment, preventing errors during the execution of daemons such as NameNode, DataNode, and ResourceManager. |
|  |
| 1  2  3  4  5  6  7  8  9  1  0  1  <?xml version="1.0">  <configuration>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <property>  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</ name>  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  </property> |
|  |
|  |
| ***Command*:** vi hadoop–env.sh |
|  |
|  |
|  |
|  |
| *Fig: Hadoop Installation – Configuring hadoop-env.sh* |
|  |
|  |
|  |
| **Step 12: Format the NameNode** Formatting the NameNode initializes the Hadoop Distributed File System (HDFS) by setting up the required filesystem structure within the directory defined by the dfs.name.dir configuration property. This step prepares the NameNode for managing metadata of the distributed filesystem.  **Note:** Formatting should be performed **only once during initial setup**. Reformatting an active HDFS will result in **irrecoverable data loss**. |
| Navigate to the Hadoop home directory: cd  cd hadoop-2.7.3 |
| Format the NameNode: bin/hadoop namenode -format |
|  |
| Once successfully formatted, a new file system structure will be created, and Hadoop is ready to start its core daemons. |
|  |
|  |
| **Fig: Hadoop Installation – Formatting NameNode** |
|  |
| **Step 13: Start Hadoop Daemons** After formatting the NameNode, the essential Hadoop daemons must be started to initiate the Hadoop Distributed File System (HDFS), YARN resource manager, and the MapReduce history server. These daemons coordinate storage, processing, and resource management across the cluster. **Navigate to the** sbin **directory:** bash  cd hadoop-2.7.3/sbin **Option A: Start All Daemons at Once** Use the following command to start all core Hadoop services simultaneously:  bash  ./start-all.sh  This command internally triggers the following individual scripts:   * start-dfs.sh – Starts HDFS daemons (NameNode, DataNode) * start-yarn.sh – Starts YARN daemons (ResourceManager, NodeManager) * mr-jobhistory-daemon.sh – Starts MapReduce JobHistory Server  **Option B: Start Services Individually** **Start NameNode:**  bash  CopyEdit  ./hadoop-daemon.sh start namenode  **Figure:** Starting the HDFS NameNode daemon  The **NameNode** is the master server of HDFS. It manages the metadata of the file system and tracks the location of blocks across DataNodes.  Other services like **DataNode**, **ResourceManager**, **NodeManager**, and **JobHistoryServer** can also be started individually using their respective commands. |
|  |
|  |
|  |
| Step 14 Start DataNode: On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.  ***Command:*** ./hadoop-daemon.sh start datanode    **Fig: Hadoop Installation – Starting DataNode** |
|  |
| Start ResourceManager: ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application’s ApplicationMaster.  ***Command:*** ./yarn-daemon.sh start resourcemanager  **Fig: Hadoop Installation – Starting ResourceManager** |
|  |
| Start NodeManager: The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.  ***Command:*** ./yarn-daemon.sh start nodemanager |
| [See Batch Details](https://www.edureka.co/big-data-hadoop-training-certification)  **Fig: Hadoop Installation – Starting NodeManager** |
| Start JobHistoryServer: JobHistoryServer is responsible for servicing all job history related requests from client.  ***Command*:** ./mr-jobhistory-daemon.sh start historyserver |
|  |
| Step 14: **To check that all the Hadoop services are up and running, run the below command.**  ***Command:*** jps  **Fig: Hadoop Installation – Checking Daemons** |
| **Step 15:** Now open the Mozilla browser and go  to **localhost**:**50070/dfshealth.html** to check the NameNode interface.  **Fig: Hadoop Installation – Starting WebUI**  Congratulations, you have successfully installed a single node Hadoop cluster |

|  |  |
| --- | --- |
| **RESULT :** | The single-node Hadoop cluster setup was completed successfully. The MapReduce-based WordCount application was executed, demonstrating the capability of Hadoop to perform distributed processing of data, even in a minimal node environment. |