## DEPARTMENT OF COMPUTER SCIENCE

#### Big Data Analytics

**Lab Manual**

****

**RATHINAM COLLEGE OF ARTS & SCIENCE**

**(AUTONOMOUS)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prepared by:** | **Approved & Reviewed by:** | **Issued by:** | **W.e.f Date:** |
| SUKANYA C K |  |  | 2025 |

**In-charge HOD**

**Principal**



#### RATHINAM COLLEGE OF ARTS & SCIENCE

Rathinam Techzone Campus, Pollachi Road, Eachanari, Coimbatore - 641021, Tamil Nadu. [**http://www.rathinamcollege.com**](http://www.rathinamtechzone.com/)

**Department of Coputer Science**

### Lab Manual for the Academic Year 2021-22 (in accordance with Computer Science syllabus)

SUBJECT : BIG DATA ANALYTICS STREAM : B.Sc [ AIDS ]

H.O.D DEAN

**List of Lab Exercises**

1. Prepare infrastructure and understand objective for software requirement for setting up

single node Hadoop cluster.

* + WinSCP
  + Putty
  + Ubuntu
  + VMPlayer
  + Hadoop version

1. Create single node Hadoop cluster.
   * Installing Ubuntu on VM
   * Installing Java
   * SSH Configuration
   * Core-site.xml Configuration
   * Hdfs-site.xml Configuration
   * Yarn-site.xml Configuration
2. Testing Single Node cluster, Web UI ports and Exploring different daemons of Hadoop Cluster.
3. Perform / Execute below sets of Hadoop basic commands:
   * appendToFile
   * cat
   * chgrp
   * chmod
   * chown
   * copyFromLocal
   * copyToLocal
   * count
   * cp
4. Install eclipse IDE on single node cluster for executing Map Reduce Job and understand the role of dependent libraries for processing job.

###### PROGRAM1

1. Prepare infrastructure and understand objective for software requirement for setting up

single node Hadoop cluster.

* + WinSCP
  + Putty
  + Ubuntu
  + VMPlayer
  + Hadoop version

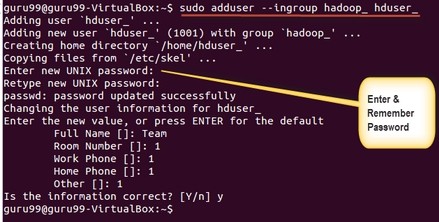
**Part 1) Download and Install Hadoop**

**Step 1)** Add a Hadoop system user using below command

sudo addgroup hadoop\_



sudo adduser --ingroup hadoop\_ hduser\_



This error can be resolved by Login as a root user



Execute the command

sudo adduser hduser\_ sudo



Re-login as hduser\_



**Step 2)** Configure SSH

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

First, switch user, enter the following command

su - hduser\_



This command will create a new key.

ssh-keygen -t rsa -P ""

Enable SSH access to local machine using this key.

cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys



Now test SSH setup by connecting to localhost as ‘hduser’ user.

ssh localhost

##### To resolve this –

Purge SSH using,

sudo apt-get purge openssh-server

It is good practice to purge before the start of installation

Install SSH using the command-

sudo apt-get install openssh-server



Step 3) Next step is to [Download Hadoop](https://www.guru99.com/install-linux.html) Select the tar.gz file ( not the file with src)



Once a download is complete, navigate to the directory containing the tar file



Part 2) Configure Hadoop

**Step 1)** Modify **~/.bashrc** file

Add following lines to end of file **~/.bashrc Step 2)** Configurations related to HDFS Set **JAVA\_HOME** inside

##### file $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

There are two parameters

in **$HADOOP\_HOME/etc/hadoop/core-site.xml** which need to be set-

1. ‘**hadoop.tmp.dir’ –** Used to specify a directory which will be used by Hadoop to store its data files.
2. **‘fs.default.name’ –** This specifies the default file system.

To set these parameters, open core-site.xml

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

**Step 3)** Map Reduce Configuration

Before you begin with these configurations, lets set HADOOP\_HOME path

sudo gedit /etc/profile.d/hadoop.sh

export HADOOP\_HOME=/home/guru99/Downloads/Hadoop sudo chmod +x /etc/profile.d/hadoop.sh

Type echo $HADOOP\_HOME. To verify the path



##### Step

**4)** Before we start Hadoop for the first time, format HDFS using below command

$HADOOP\_HOME/bin/hdfs namenode -format

**Step 5)** Start Hadoop single node cluster using below command

$HADOOP\_HOME/sbin/start-dfs.sh.

**Step 6)** Stopping Hadoop

$HADOOP\_HOME/sbin/stop-dfs.sh

## PROGRAM2

1. Create single node Hadoop cluster.
   * Installing Ubuntu on VM
   * Installing Java
   * SSH Configuration
   * Core-site.xml Configuration
   * Hdfs-site.xml Configuration
   * Yarn-site.xml Configuration

STPES FOR INSTALLATION:

First, we need to install SSH and few software installation utilities for Java 8:

sudo apt install

openssh-server

software-properties-common

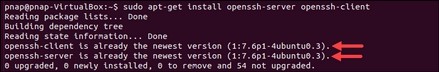
python-software-properties

# Install OpenJDK on Ubuntu

sudo apt update

sudo apt install openjdk-8-jdk -y java -version; javac -version

sudo apt install openssh-server openssh-client -y



### Create Hadoop User

sudo adduser hdoop su – hdoop

### Enable Passwordless SSH for Hadoop User

[Generate an SSH key pair](https://phoenixnap.com/kb/generate-setup-ssh-key-ubuntu) and define the location is is to be stored in:

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

Use the **cat** command to store the public key as **authorized\_keys** in the ssh directory:

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

Set the permissions for your user with the **chmod** command:

chmod 0600 ~/.ssh/authorized\_keys

The new user is now able to SSH without needing to enter a password every time. Verify everything is set up correctly by using the **hdoop** user to SSH to localhost:

ssh localhost

Use the provided mirror link and download the Hadoop package with the **wget** command:

wget

https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop

-3.2.1.tar.gz

Once the download is complete, extract the files to initiate the Hadoop installation:

tar xzf hadoop-3.2.1.tar.gz

### Configure Hadoop Environment Variables (bashrc)

Edit the .bashrc shell configuration file using a text editor of your choice (we will be using nano):

sudo nano .bashrc

Define the Hadoop environment variables by adding the following content to the end of the 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/nativ"

### Edit hadoop-env.sh File

The hadoop-env.sh file serves as a master file to configure YARN, [HDFS](https://phoenixnap.com/kb/what-is-hdfs), [MapReduce](https://phoenixnap.com/kb/hadoop-mapreduce), and Hadoop-related project settings.

sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64 which javac

readlink -f /usr/bin/javac

### Edit mapred-site.xml File

Use the following command to access the mapred-site.xml file and **define MapReduce values**:

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

### Edit yarn-site.xml File

The yarn-site.xml file is used to define settings relevant to **YARN**. It contains configurations for the **Node Manager, Resource Manager, Containers,** and **Application Master**.

Open the yarn-site.xml file in a text editor:

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

Format HDFS NameNode

It is important to **format the NameNode** before starting Hadoop services for the first time:

hdfs namenode -format

Start Hadoop Cluster

Navigate to the hadoop-3.2.1/sbin directory and execute the following commands to start the NameNode and DataNode:

./start-dfs.sh

Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

./start-yarn.sh

## PROGRAM3

1. **Testing Single Node cluster, Web UI ports and Exploring different daemons of Hadoop Cluster.**

**Start NameNode:**

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start namenode

**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

**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

**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

**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

###### PROGRAM4

Perform / Execute below sets of Hadoop basic commands:

* appendToFile
* cat
* chgrp
* chmod
* chown
* copyFromLocal
* copyToLocal
* count
* cp

appendToFile

hdfs dfs -appendToFile <localsrc> ... <dst>

cat

Usage: hdfs dfs -cat URI [URI …]

chgrp

Usage: hdfs dfs -chgrp [-R] GROUP URI [URI …]

chmod

Usage: hdfs dfs -chmod [-R] <MODE[,MODE]... | OCTALMODE> URI [URI ...]

chown

Usage: hdfs dfs -chown [-R] [OWNER][:[GROUP]] URI [URI ]

copyFromLocal

Usage: hdfs dfs -copyFromLocal <localsrc> URI

copyToLocal

Usage: hdfs dfs -copyToLocal [-ignorecrc] [-crc] URI

<localdst>

count

Usage: hdfs dfs -count [-q] <paths>

cp

Usage: hdfs dfs -cp [-f] URI [URI ...] <dest>

du

Usage: hdfs dfs -du [-s] [-h] URI [URI ...]

###### PROGRAM5

**5. Install eclipse IDE on single node cluster for executing Map Reduce Job and understand the role of dependent libraries for processing job**

**Step 1.** Launch Eclipse and set the Eclipse Workspace.

**Step 2.** To create the Hadoop MapReduce Project, click on **File**

##### >> New >> Java Project.

**Step 3.** Create a new Package **right-click on the Project Name >> New >> Package**.

**Step 4.** Add the Hadoop libraries (jars).

**Step 5.** Now create a new class that performs the map job. Here in this article, **WordCountMapper** is the class for performing the mapping task.

##### Right-Click on Package Name >> New >> Class

**Step 6.** Copy the below code in your class created above for the mapper.

Press **Ctrl+S** to save the code.

**Step 7.** Now create another class (in the same way as we used above), for creating a class that performs the reduce job.

Here in this article, **WordCountReducer** is the class to perform the reduce task.

Click **Finish**.

**Step 8.** Copy the below code in your class created above for the reducer.

Press **Ctrl+S** to save the code.

**Step 9.** Now create the driver class, which contains the main method. Here in this article, the driver class for the project is named “**WordCount”**.

Click **Finish**.

**Step 10.** Copy the below code in your driver class, which contains the main method.

Press **Ctrl+S** to save the Code.

**Step 11.** Creating the **Jar File** of the Project

Before running created Hadoop MapReduce word count application, we have to create a jar file.

To do so **Right-click on project name >> Export**. Select the **JAR file** option. Click Next.

Provide the **Jar file name**:

Click **Finish**.

**Step 12.** Execute the Hadoop MapReduce word count application using the below execution command. **hadoop jar <project jar file path> <input file path>**

##### <output directory>

**tep 13.** Browse the Hadoop MapReduce Word Count Project Output.

The output directory of the Project in HDFS contains two files:

\_SUCCESS and part-r-00000

The output is present in the /part-r-00000 file.

You can browse the result using the below command.

**hadoop fs -cat <output directory/part-r-00000>**