# Exp No: 9 Date: 09/10/2024

**HADOOP**

**SET UP A SINGLE HADOOP CLUSTER AND SHOW THE PROCESS USING WEB UI**

**AIM:**

To set-up one node Hadoop cluster.

# PROCEDURE:

1. System Update
2. Install Java
3. Add a dedicated Hadoop user
4. Install SSH and setup SSH certificates
5. Check if SSH works
6. Install Hadoop
7. Modify Hadoop config files
8. Format Hadoop filesystem
9. Start Hadoop
10. Check Hadoop through web UI
11. Stop Hadoop

# THEORY:

Hadoop is an Apache open-source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from a single server to thousands of machines, each offering local computation and storage.

# HADOOP ARCHITECTURE

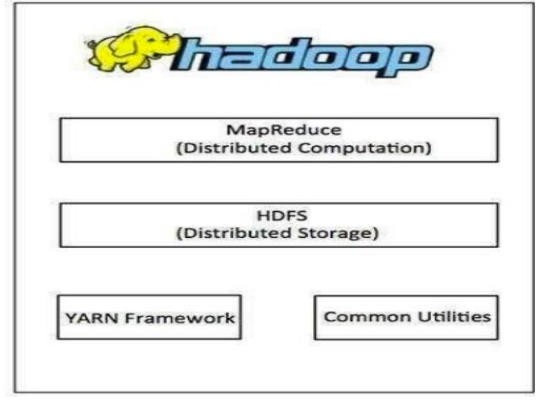
Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS): A distributed file system that provides high- throughput access to application data.

Hadoop MapReduce: This is a YARN-based system for parallel processing of large data sets. We can use following diagram to depict these four components available in Hadoop framework.



# PROCEDURE:

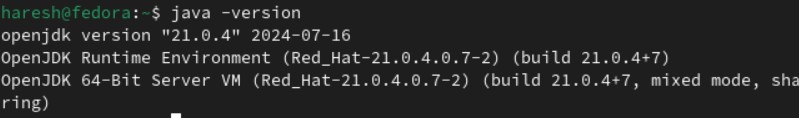
**Step 1 – System Update**

$ sudo apt-get update

# Step 2 – Install Java and Set JAVA\_HOME

$ sudo apt-get install default-jdk

$ java -version



# Step 3 – Add a dedicated Hadoop user

$ sudo addgroup Hadoop

$ sudo adduser --ingroup hadoop hduser

// Add hduser to sudo user group

$ sudo adduser hduser sudo

# Step 4 – Install SSH and Create Certificates

$ sudo apt-get install ssh

$ su hduser

$ ssh-keygen -t rsa -P ""

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

# Step 6 – Install Hadoop

$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.8.4/hadoop-2.8.4.tar.gz

// Extract Hadoop-2.8.4

$ sudo tar xvzf hadoop-2.8.4.tar.gz



// Create a folder ‘hadoop’ in /usr/local

$ sudo mkdir –p /usr/local/Hadoop

// Move the Hadoop folder to /usr/local/hadoop

$ sudo mv hadoop-2.8.4 /usr/local/Hadoop

// Assigning read and write access to Hadoop folder

$ sudo chown –R hduser:hadoop /usr/local/Hadoop

# Step 7 - Modify Hadoop config files

//Hadoop Environmental variable setting – The following files will be modified

1. ~/.bashrc
2. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/core-site.xml
4. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hdfs-site.xml
5. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/yarn-site.xml
6. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml.template

$ sudo nano ~/.bashrc

// Add the following lines at the end of the file exportJAVA\_HOME=/usr/lib/jvm/java-8-oracle

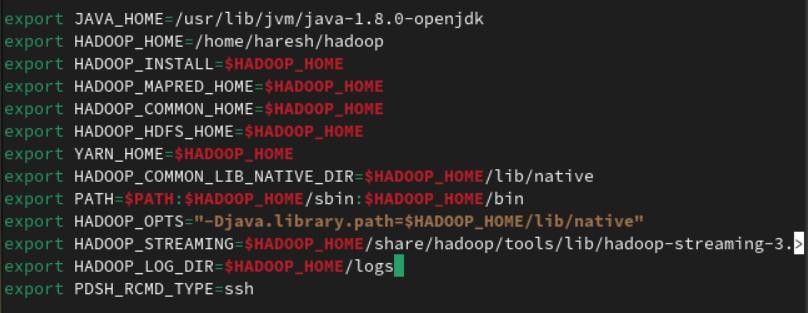
export HADOOP\_HOME=/usr/local/hadoop/hadoop-2.8.4 export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME export HADOOP\_COMMON\_HOME=$HADOOP\_HOME export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native export HADOOP\_OPTS="-D.java.library.path=$HADOOP\_HOME/lib" export PATH=$PATH:/usr/local/hadoop/hadoop-2.8.4/bin



// Configure Hadoop Files

$ cd /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/

$ sudo nano hadoop-env.sh

// Add following line in hadoop-env.sh – Set JAVA variable in Hadoop # The java implementation to use.

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



// Create datanode and namenode

$ sudo mkdir –p /usr/local/hadoop\_tmp/hdfs/namenode

$ sudo mkdir –p /usr/local/hadoop\_tmp/hdfs/datanode

// Changing ownership to hadoop\_tmp

$ sudo chown –R hduser:hadoop /usr/local/hadoop\_tmp

// Edit hdfs-site.xml

$ sudo nano hdfs-site.xml

// Add the following lines between <configuration> </configuration>

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop\_tmp/hdfs/namenode</value>

</property>

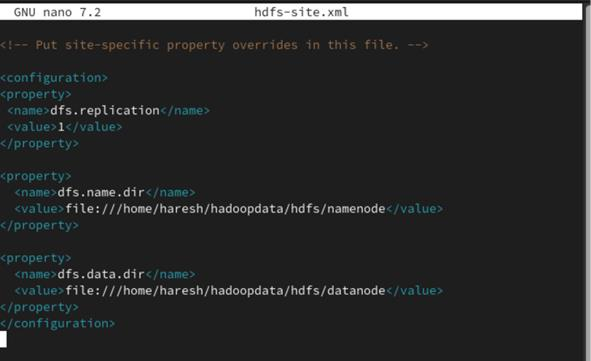
<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop\_tmp/hdfs/datanode</value>

</property>

</configuration>



// Edit core-site.xml

$ sudo nano core-site.xml

21

// Add the following lines between <configuration> </configuration>

<configuration>

<property>

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

<value>hdfs://localhost:9000</value>

</property>

</configuration>

// Edit yarn-site.xml

$ sudo nano yarn-site.xml

// Add the following lines between <configuration> </configuration>

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

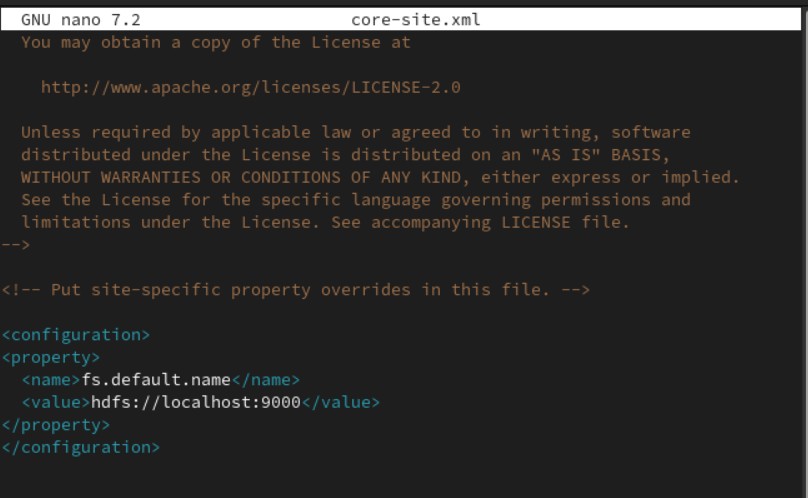
<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.Shuffle-Handler</value>

</property>

</configuration>



// Edit mapred-site.xml

$ cp /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml.template

/usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml

$ sudo nano mapred-site.xml

// Add the following lines between <configuration> </configuration>

<configuration>

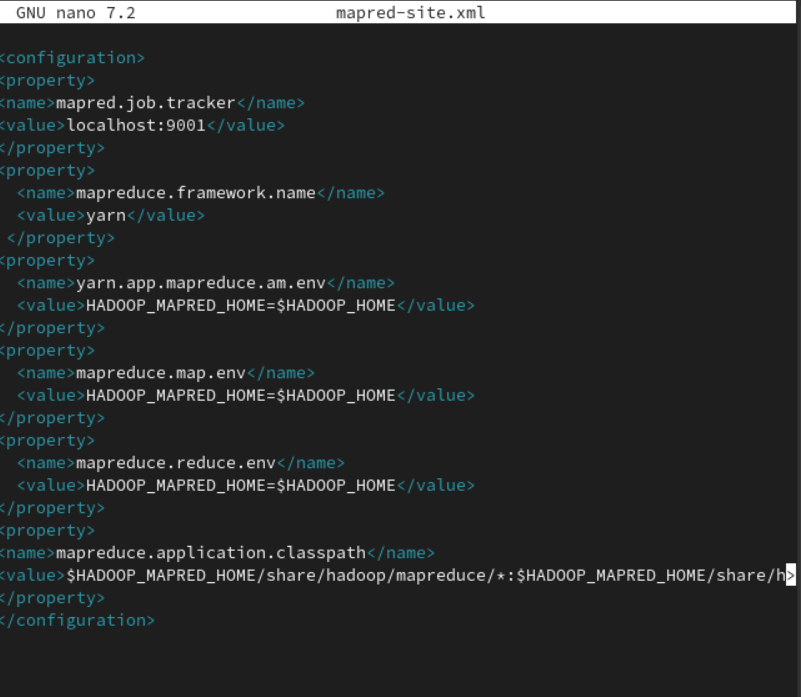
<property>

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

<value>yarn</value>

</property>

</configuration>



# Step 8 – Format Hadoop File System

$ cd /usr/local/hadoop/hadoop-2.8.4/bin

$ hadoop namenode -format

# Step 9 - Start Hadoop

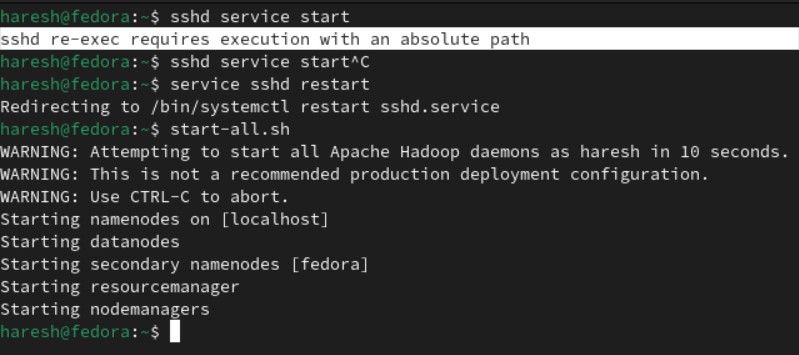
$ cd /usr/local/hadoop/hadoop-2.8.4/sbin

// Starting dfs services

$ start-dfs.sh

// Starting mapreduce services

$ start-yarn.sh

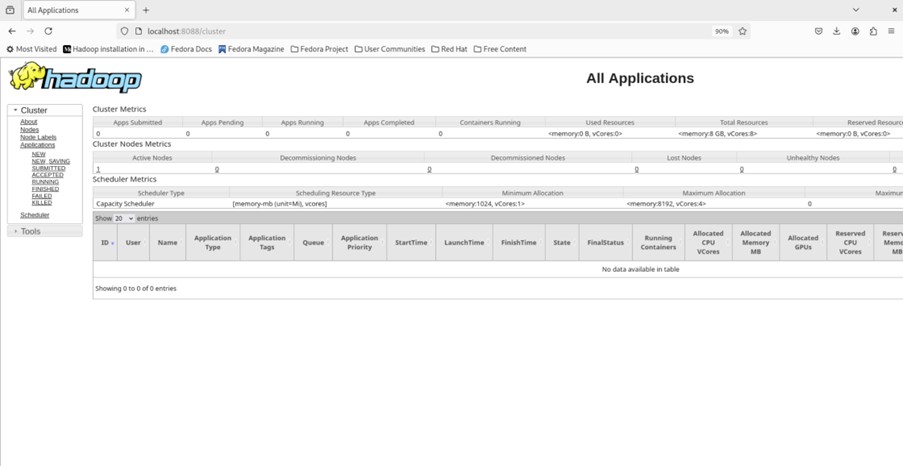


$ jps

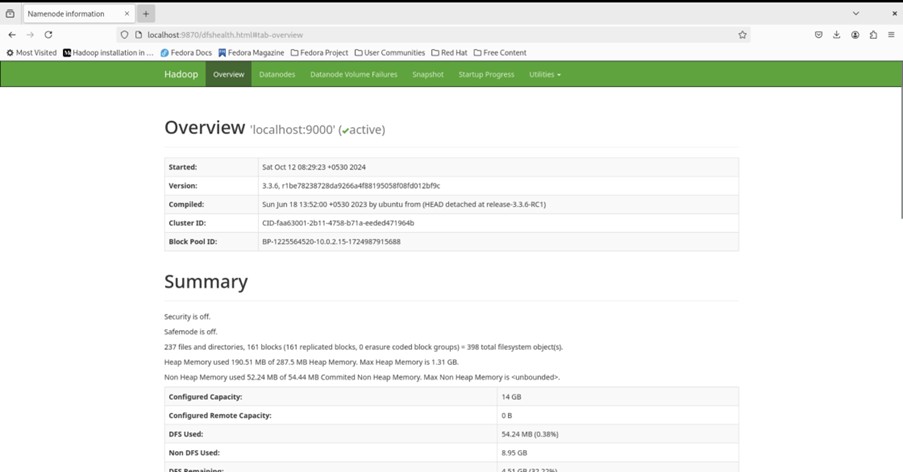


# Step 10 - Check Hadoop through web UI

Go to browser type http://localhost:8088 – All Applications Hadoop Cluster



Go to browser type http://localhost:50070 – Hadoop Namenode



# Step 11 - Stop Hadoop

$ stop-dfs.sh

$ stop-yarn.sh

# RESULT:

Hadoop has been successfully installed and nodes have been successfully created.