Exp No: 9 Date:

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.

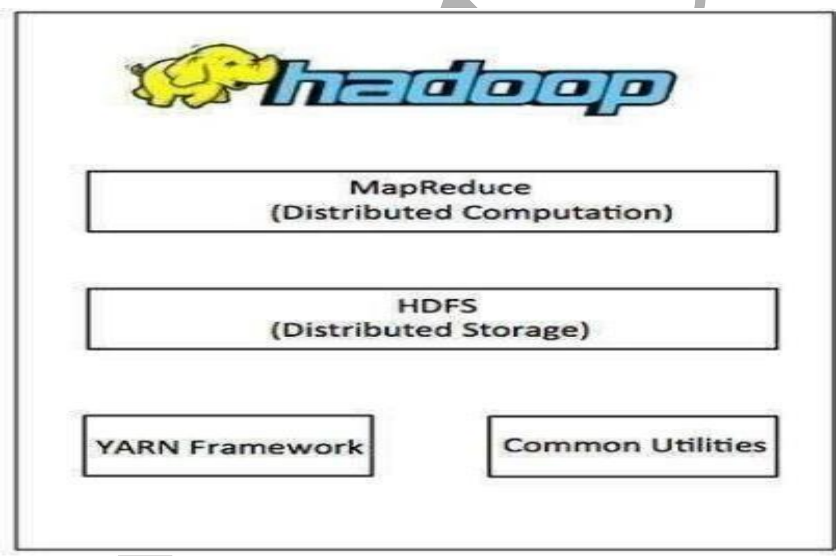
Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

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

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

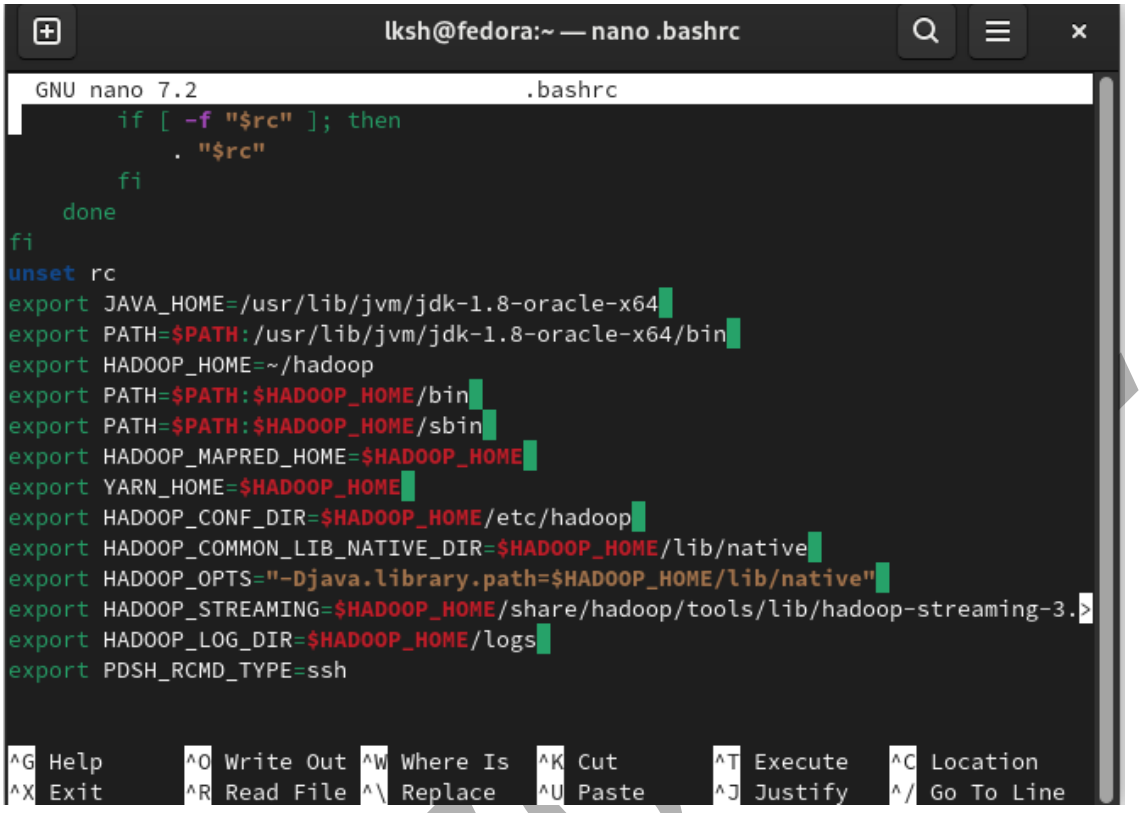
Hadoop MapReduce: This is a YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.

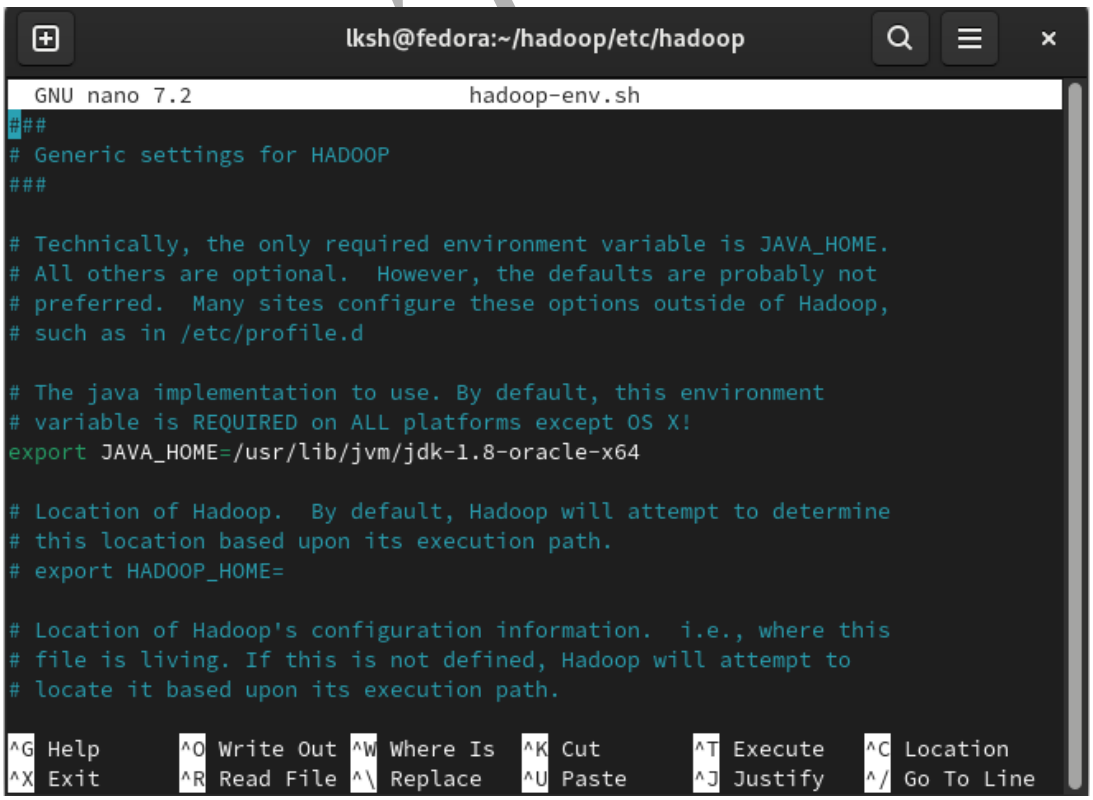


PROCEDURE:

$ nano ~/.bashrc

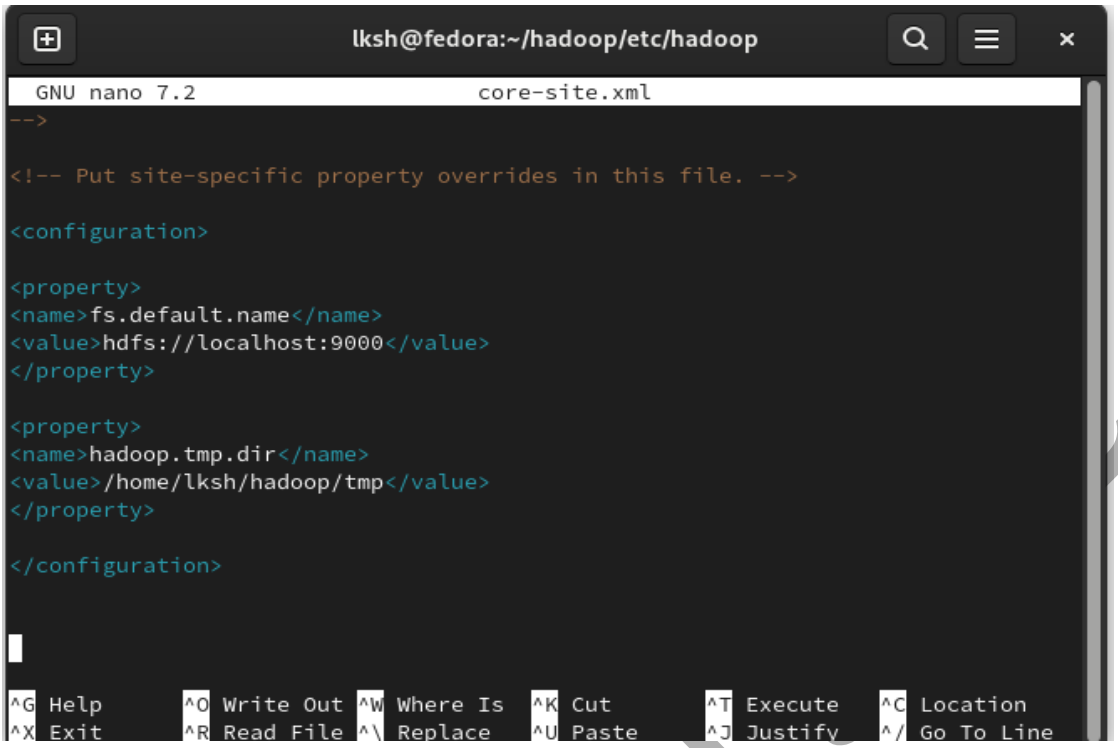


$ nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh



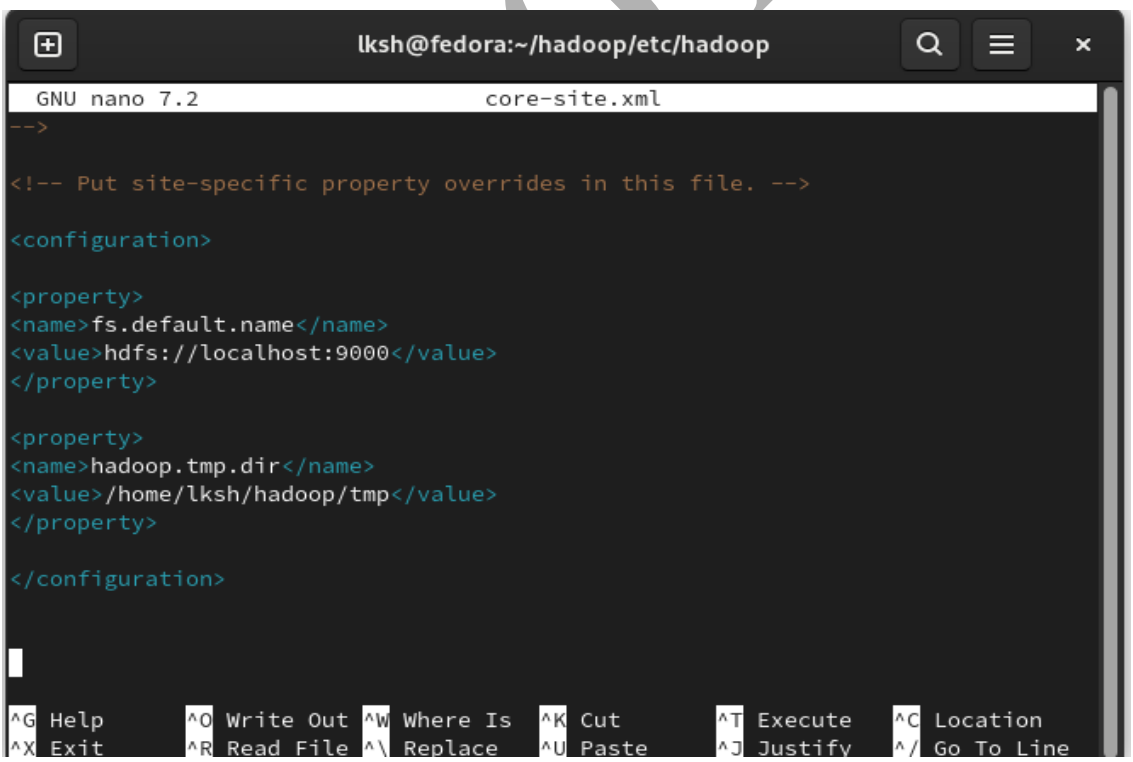
NAME:RAJA SEKAR M

$nano $HADOOP_HOME/etc/hadoop/core-site.xml



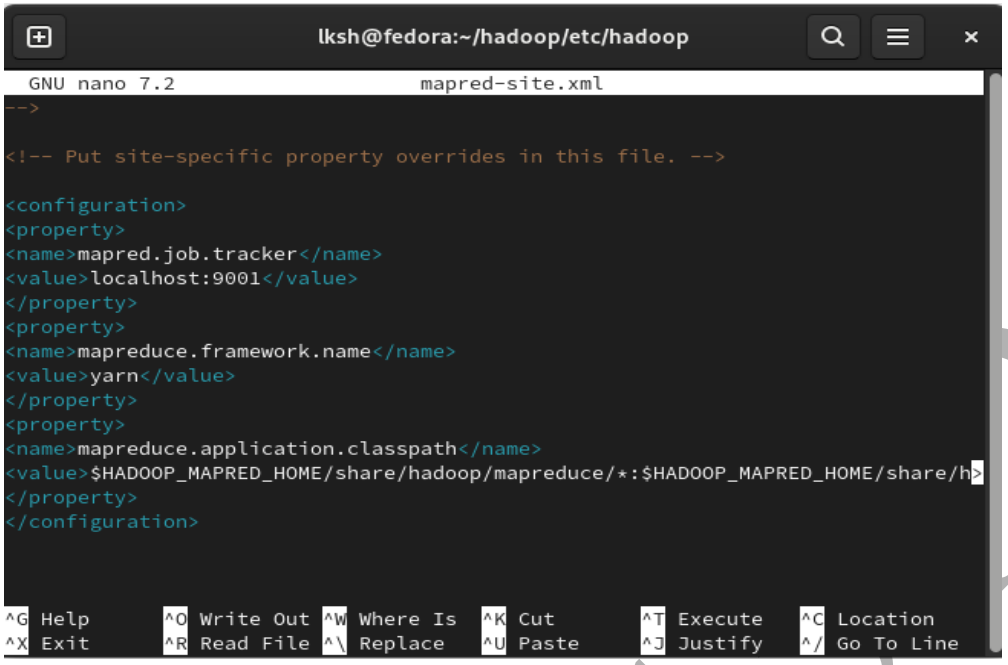

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>

<property>
<name>hadoop.tmp.dir</name>
<value>/home/lksh/hadoop/tmp</value>
</property>

</configuration>
```


$nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml



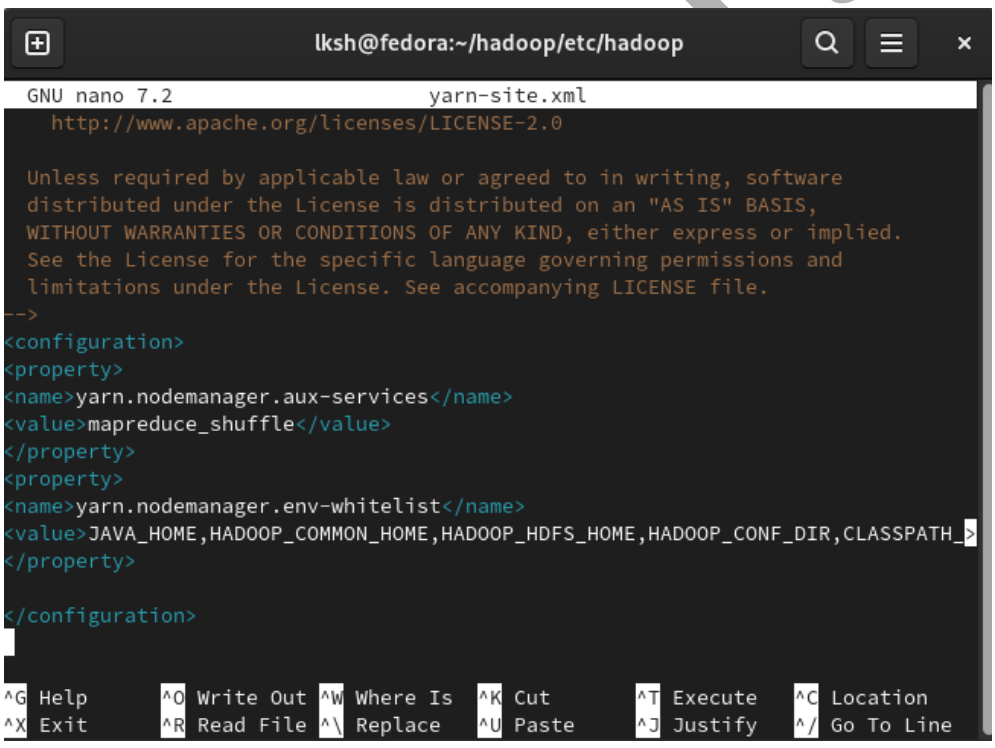

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>

<property>
<name>hadoop.tmp.dir</name>
<value>/home/lksh/hadoop/tmp</value>
</property>

</configuration>
```

$nano $HADOOP_HOME/etc/hadoop/mapred-site.xml




```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>
</property>
<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/h
</property>
</configuration>
```


$nano $HADOOP_HOME/etc/hadoop/yarn-site.xml




```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_
</property>

</configuration>
```


NAME:RAJA SEKAR M

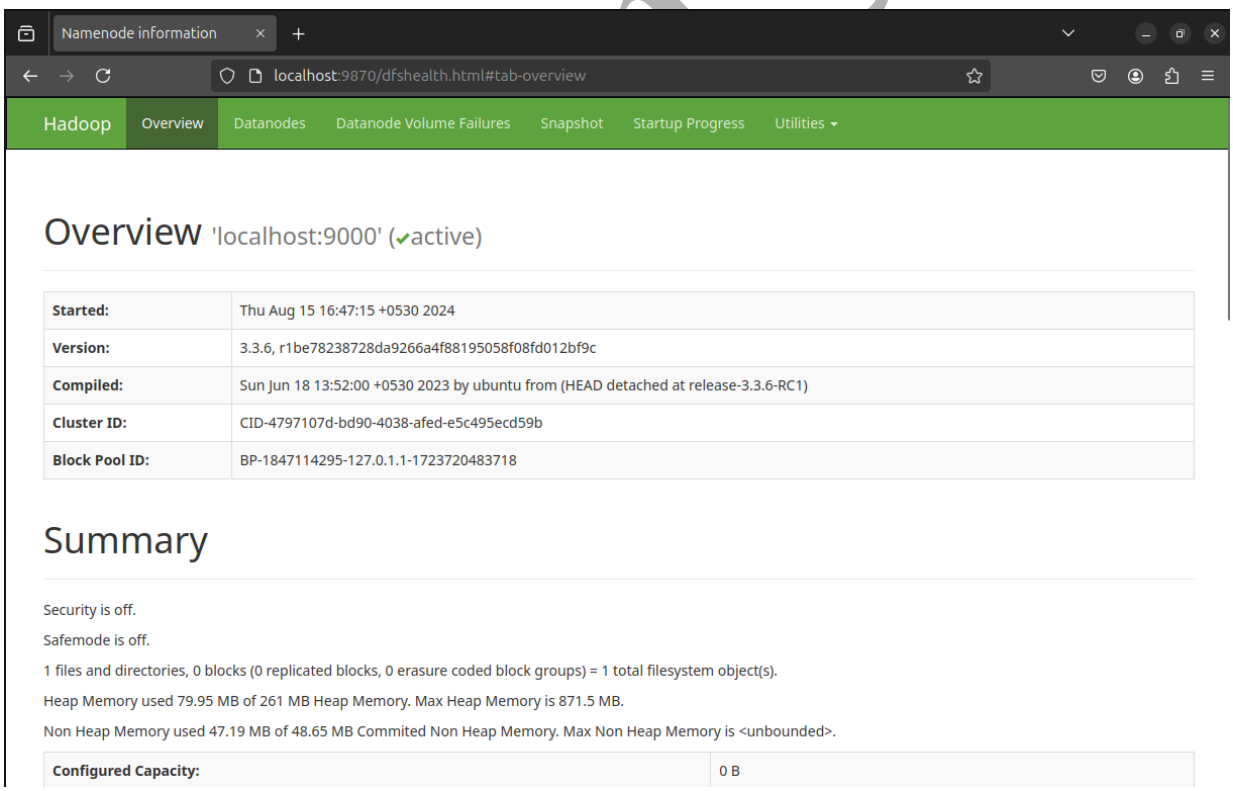
$ start-all.sh

```
lksh@fedora:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as lksh in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [fedora]
Starting resourcemanager
Starting nodemanagers
```

$ jps

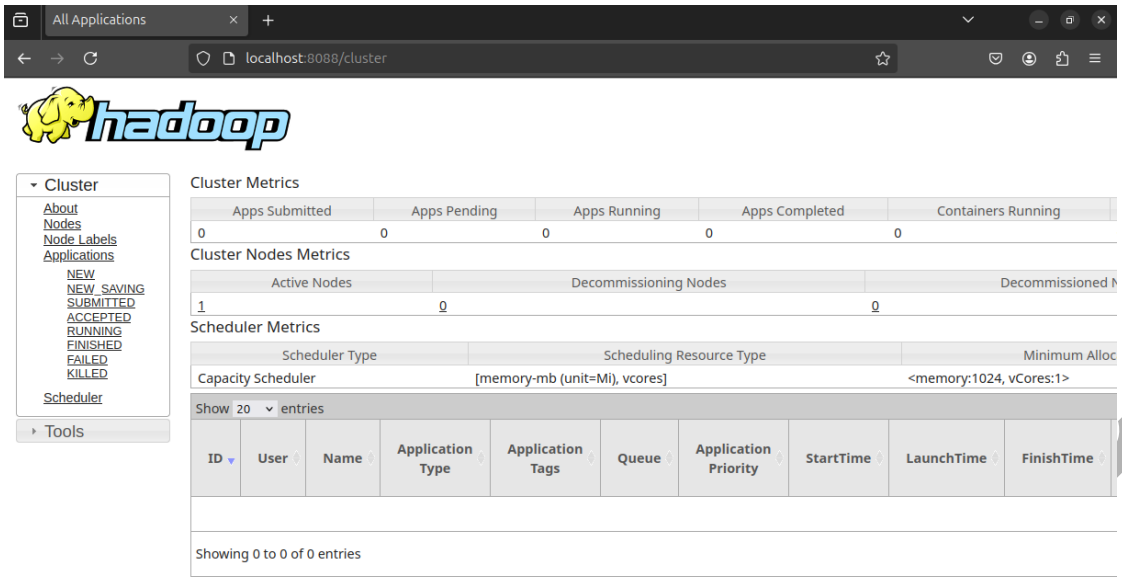
```
lksh@fedora:~/hadoop/etc/hadoop$ jps
3490 DataNode
4163 NodeManager
4599 Jps
3321 NameNode
4027 ResourceManager
3695 SecondaryNameNode
```

localhost:9870



NAME:RAJA SEKAR M

localhost:8088



RESULT:

Thus, Hadoop has been successfully installed.

NAME:RAJA SEKAR M