

**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:

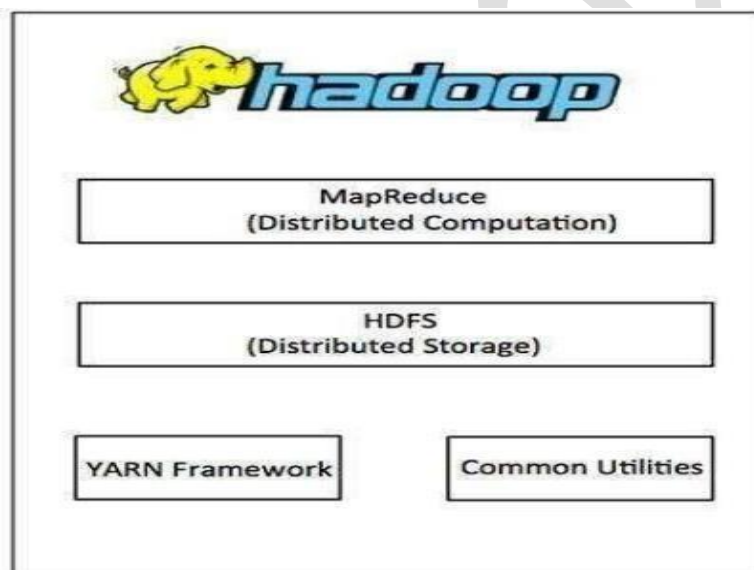
**Hadoop Common:** These are Java libraries and utilities required by other Hadoop modules. These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

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

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

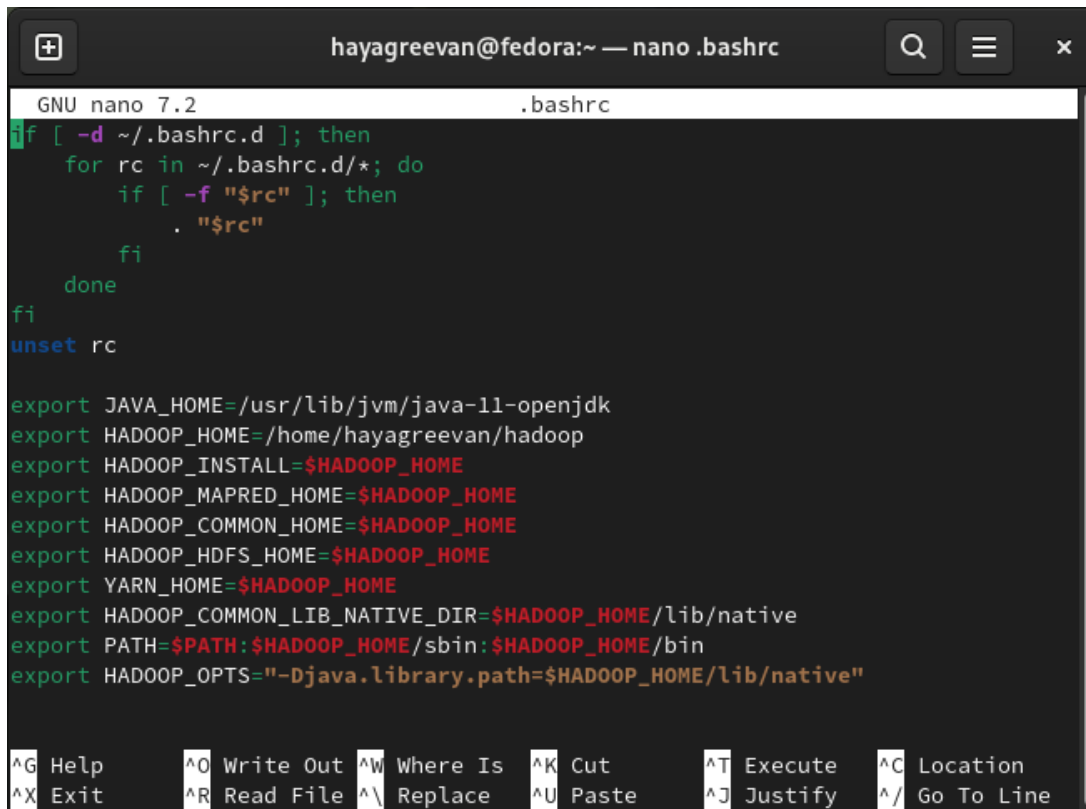
**Hadoop MapReduce:** This is a YARN-based system for parallel processing of large data sets.

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



## PROCEDURE

\$ nano ~/.bashrc

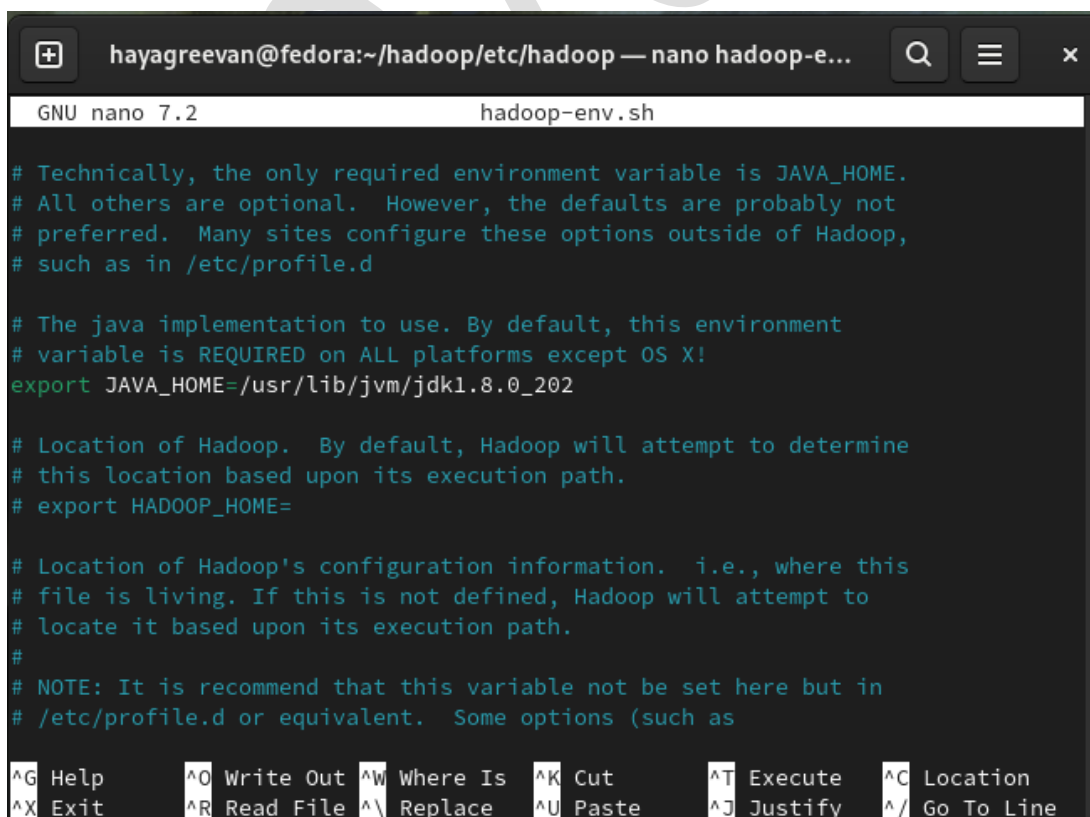


```
GNU nano 7.2                                .bashrc
if [ -d ~/.bashrc.d ]; then
  for rc in ~/.bashrc.d/*; do
    if [ -f "$rc" ]; then
      . "$rc"
    fi
  done
fi
unset rc

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk
export HADOOP_HOME=/home/hayagreevan/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:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^\_ Go To Line

\$ nano \$HADOOP\_HOME/etc/hadoop/hadoop-env.sh



```
GNU nano 7.2                                hadoop-env.sh

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

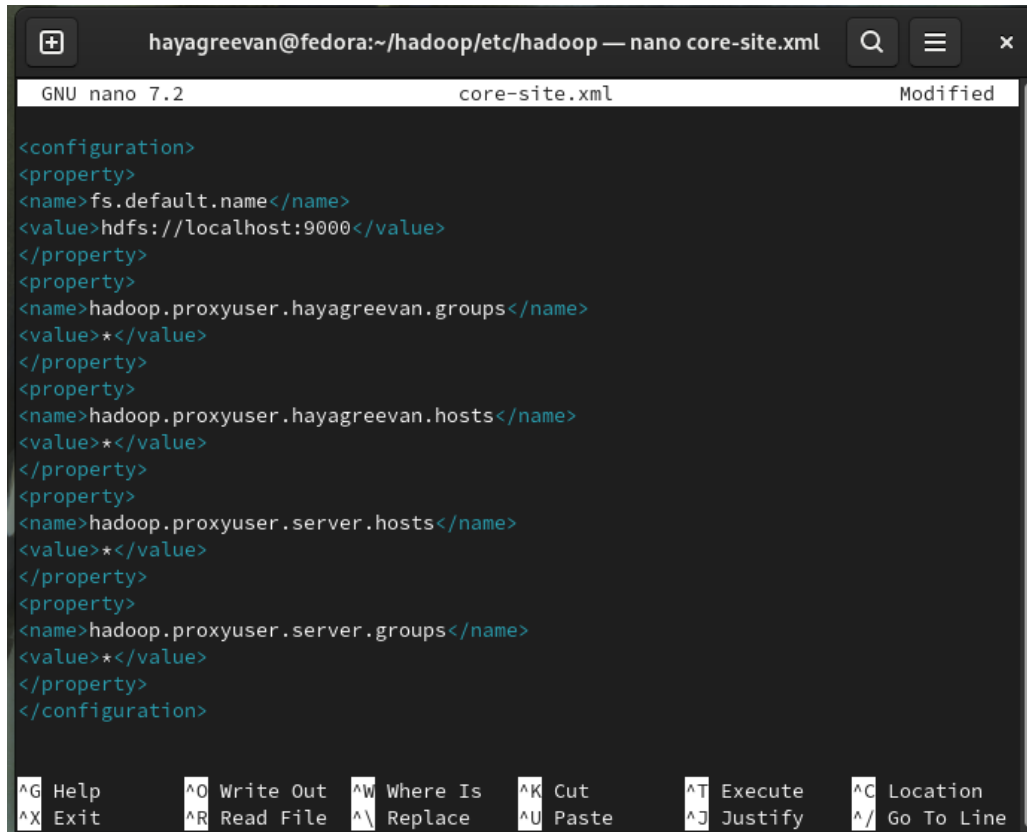
# The java implementation to use.  By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_202

# Location of Hadoop.  By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information.  i.e., where this
# file is living.  If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent.  Some options (such as
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^\_ Go To Line

**\$nano \$HADOOP\_HOME/etc/hadoop/core-site.xml**

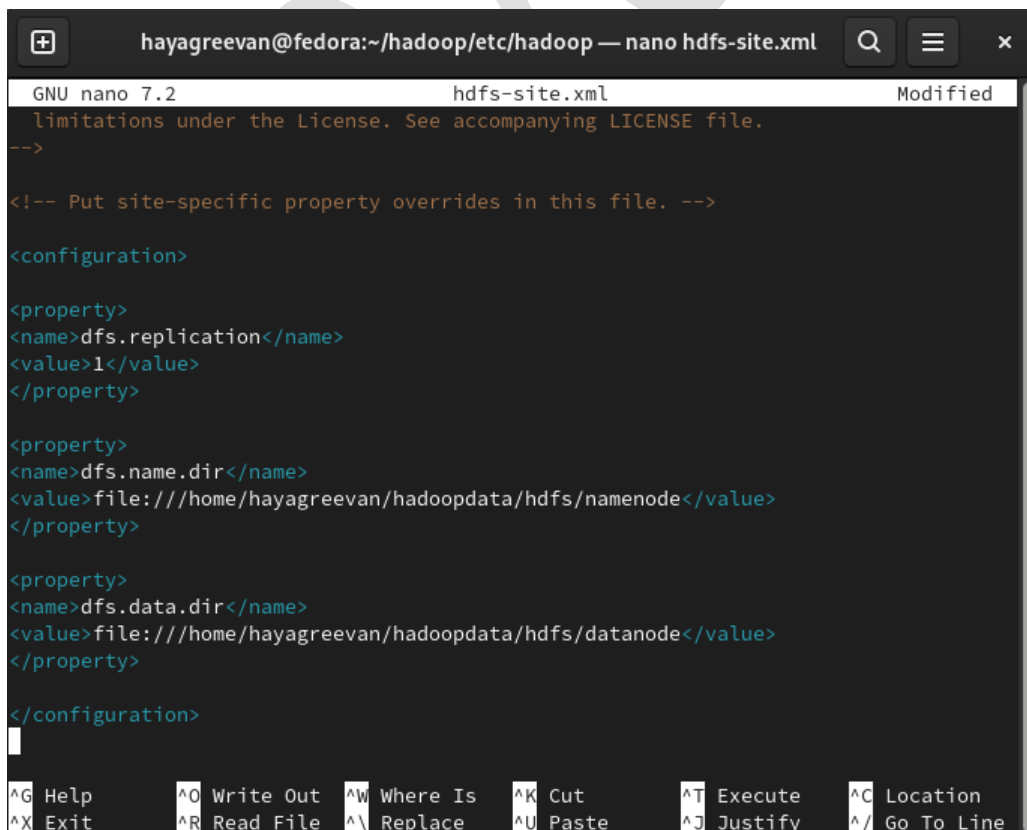


```
GNU nano 7.2 core-site.xml Modified

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
<property>
<name>hadoop.proxyuser.hayagreevan.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hayagreevan.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.server.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.server.groups</name>
<value>*</value>
</property>
</configuration>

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

**\$nano \$HADOOP\_HOME/etc/hadoop/hdfs-site.xml**



```
GNU nano 7.2 hdfs-site.xml Modified
limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>

<property>
<name>dfs.replication</name>
<value>1</value>
</property>

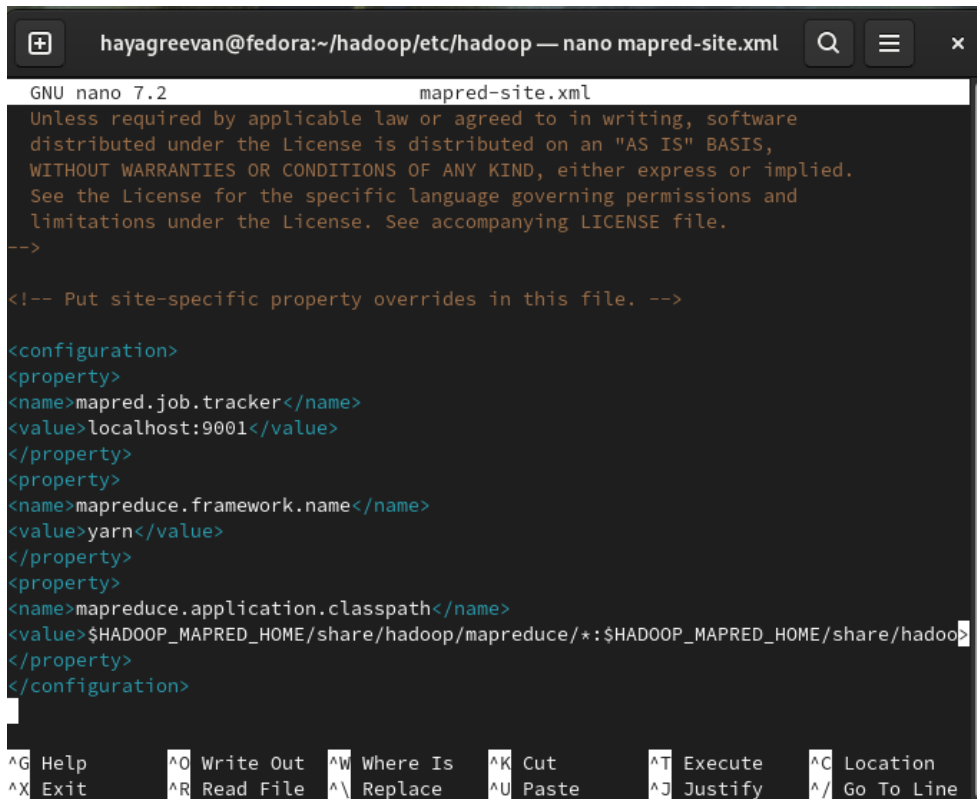
<property>
<name>dfs.name.dir</name>
<value>file:///home/hayagreevan/hadoopdata/hdfs/namenode</value>
</property>

<property>
<name>dfs.data.dir</name>
<value>file:///home/hayagreevan/hadoopdata/hdfs/datanode</value>
</property>

</configuration>

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

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



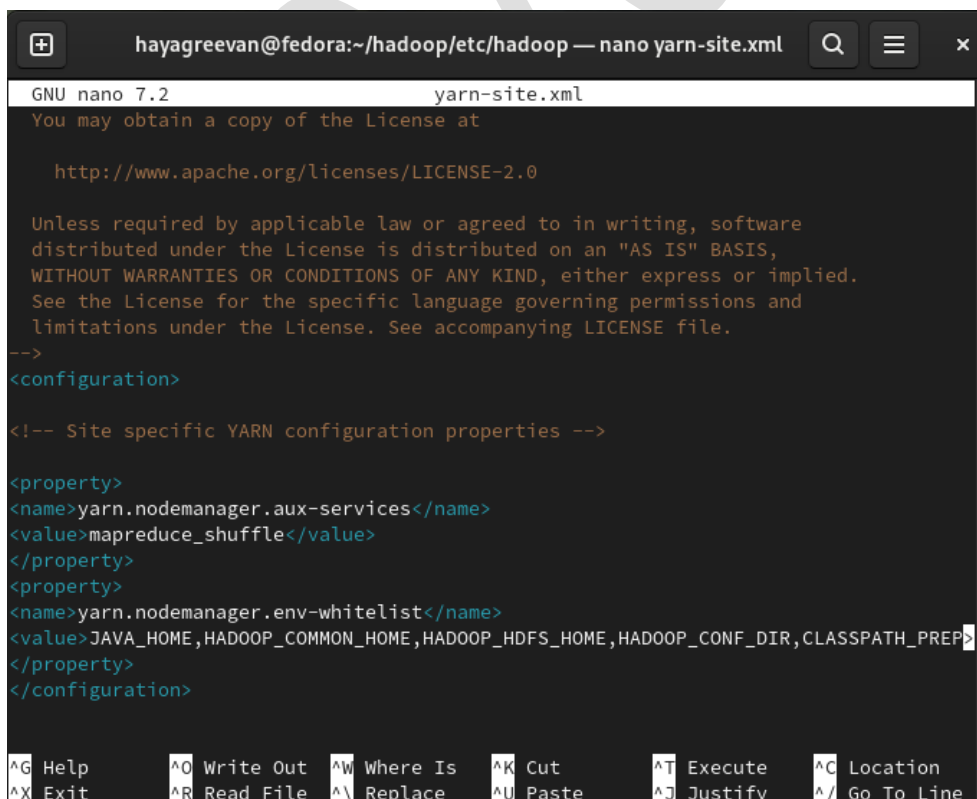
```
GNU nano 7.2 mapred-site.xml
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- 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/hadoop
</property>
</configuration>

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

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



```
GNU nano 7.2 yarn-site.xml
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<configuration>

<!-- Site specific YARN configuration properties -->

<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_PREP
</property>
</configuration>

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

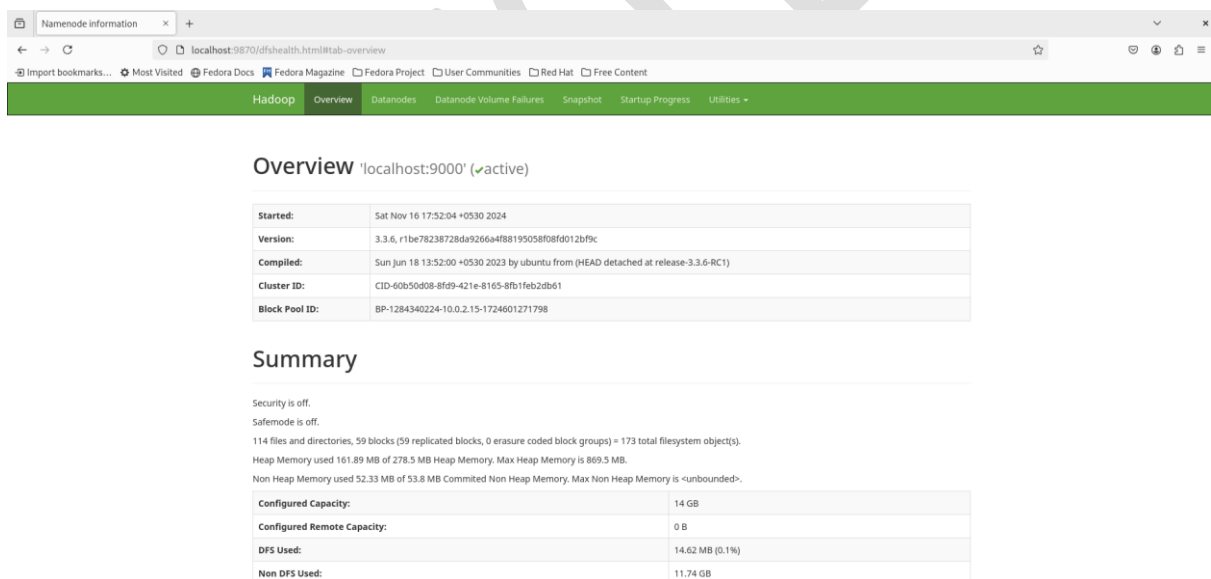
\$ start-all.sh

```
hayagreevan@fedora:~/hadoop/etc/hadoop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hayagreevan 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
hayagreevan@fedora:~/hadoop/etc/hadoop$
```

\$ jps

```
hayagreevan@fedora:~/hadoop/etc/hadoop$ jps
4225 ResourceManager
4417 NodeManager
3720 DataNode
4840 Jps
3561 NameNode
3947 SecondaryNameNode
hayagreevan@fedora:~/hadoop/etc/hadoop$
```

localhost:9870



The screenshot shows a web browser window displaying the Hadoop NameNode web interface. The browser's address bar shows the URL `localhost:9870/dfshealth.html#tab=overview`. The page has a green header with navigation tabs: **Hadoop**, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled "Overview 'localhost:9000' (✓active)". It contains a table with the following information:

|                |  |
|----------------|--|
| Started:       | Sat Nov 16 17:52:04 +0530 2024   |
| Version:       | 3.3.6, r1be78238728da9266a4f8b195058f08d012bf9c                                    |
| Compiled:      | Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1) |
| Cluster ID:    | CID-60b50d08-8fd9-421e-8165-8fb1feb2db61   |
| Block Pool ID: | BP-1284340224-10.0.2.15-1724601271798  |

Below the table is a "Summary" section. It states: "Security is off." and "Safemode is off." It also provides file and directory statistics: "114 files and directories, 59 blocks (59 replicated blocks, 0 erasure coded block groups) = 173 total filesystem object(s)." Memory usage is detailed: "Heap Memory used 161.89 MB of 278.5 MB Heap Memory. Max Heap Memory is 869.5 MB." and "Non Heap Memory used 52.33 MB of 53.8 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>." At the bottom, there is another table showing capacity and usage:

|                             |                 |
|-----------------------------|-----------------|
| Configured Capacity:        | 14 GB           |
| Configured Remote Capacity: | 0 B             |
| DFS Used:                   | 14.62 MB (0.1%) |
| Non DFS Used:               | 11.74 GB        |

**localhost:8088**

The screenshot shows the Hadoop Admin UI for a cluster named 'localhost:8088/cluster'. The page is titled 'All Applications'. On the left, there is a sidebar with navigation links: Cluster, About Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler. The main content area displays several metrics:

- Cluster Metrics:** A table showing the status of applications (Submitted, Pending, Running, Completed, Containers Running) and resource usage (Used Resources, Total Resources).
- Cluster Nodes Metrics:** A table showing the status of nodes (Active Nodes, Decommissioning Nodes, Decommissioned Nodes, Lost Nodes, Unhealthy Nodes).
- Scheduler Metrics:** A table showing the status of the scheduler (Capacity Scheduler, Scheduler Type, Scheduling Resource Type, Minimum Allocation, Maximum Allocation).
- Applications Table:** A table with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU Vcores, Allocated Memory MB, Allocated GPUs, and Reserved CPU Vcores. The table is currently empty, showing 'Showing 0 to 0 of 0 entries'.

**RESULT:**

Thus, Hadoop has been successfully installed.