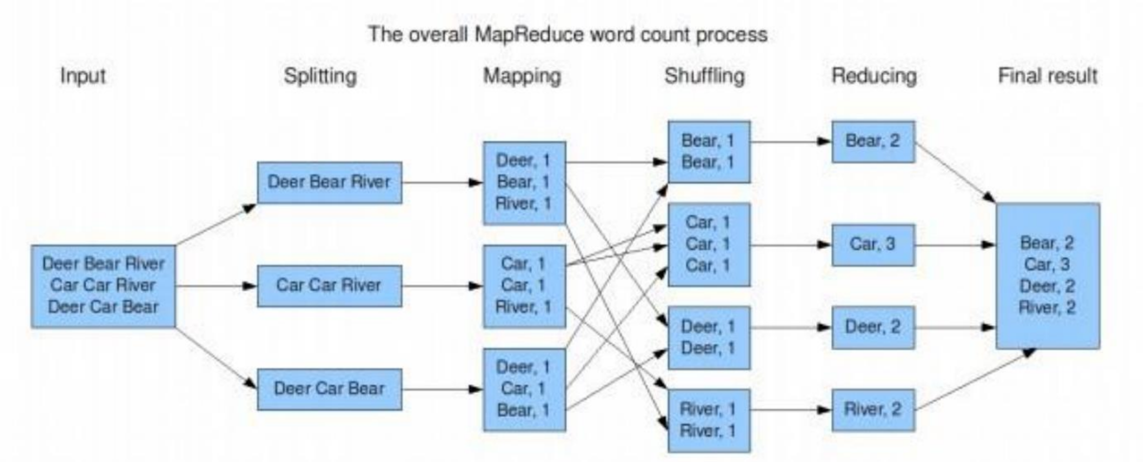
Experiment No: 07

Title: Installing Hadoop and implement program using MapReduce.

Aim: To install Hadoop and implement program using MapReduce.

Theory:

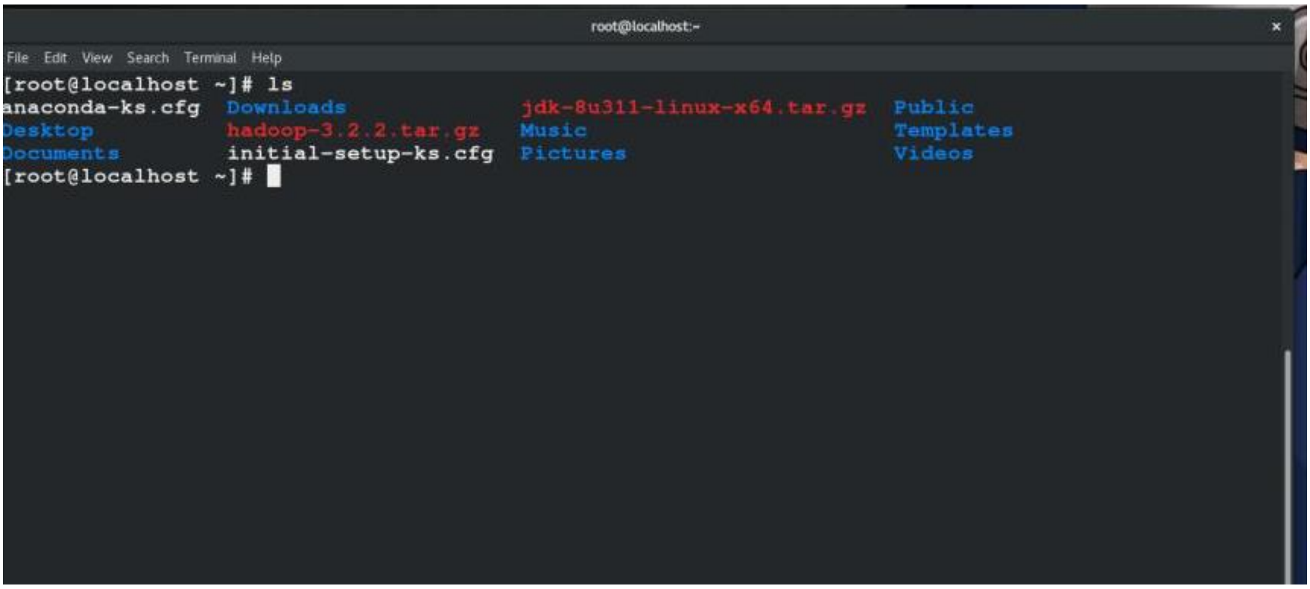
Map-Reduce is a programming model that is mainly divided into two phases:
Map Phase and Reduce Phase.
It is designed for processing the data in parallel which is divided on various machines(nodes). The Hadoop Java programs consist of Mapper class and Reducer class along with the driver class. Hadoop Mapper is a function or task which is used to process all input records from a file and generate the output which works as input for Reducer. It produces the output by returning new keyvalue pairs.
The input data has to be converted to key-value pairs as Mapper can not process the raw input records or tuples(key-value pairs). The mapper also generates some small blocks of data while processing the input records as a key-value pair. we will discuss the various process that occurs in Mapper, There key features and how the key-value pairs are generated in the Mapper.
In MapReduce word count example, we find out the frequency of each word. Here, the role of Mapper is to map the keys to the existing values and the role of Reducer is to aggregate the keys of common values.
Example :-



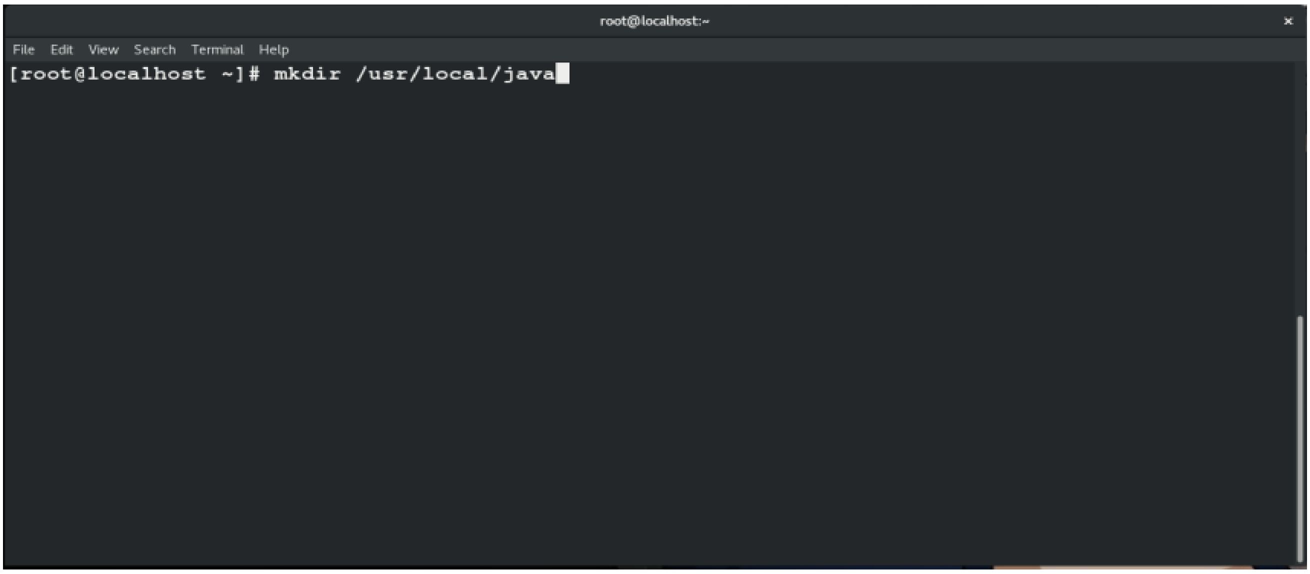
Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

At the moment, Apache Hadoop 3.x fully supports Java 8. The OpenJDK 8 package in RedHat 8 contains both the runtime environment and development kit. Move jdk from desktop to /usr/local/java

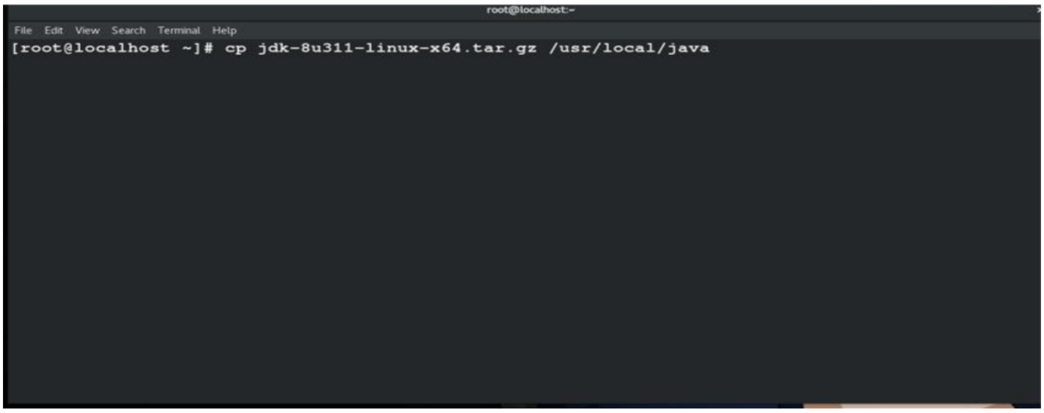
Practical:



```
[root@localhost ~]# ls
anaconda-ks.cfg  Downloads           jdk-8u311-linux-x64.tar.gz  Public
Desktop          hadoop-3.2.2.tar.gz  Music                      Templates
Documents        initial-setup-ks.cfg Pictures                   Videos
[root@localhost ~]#
```

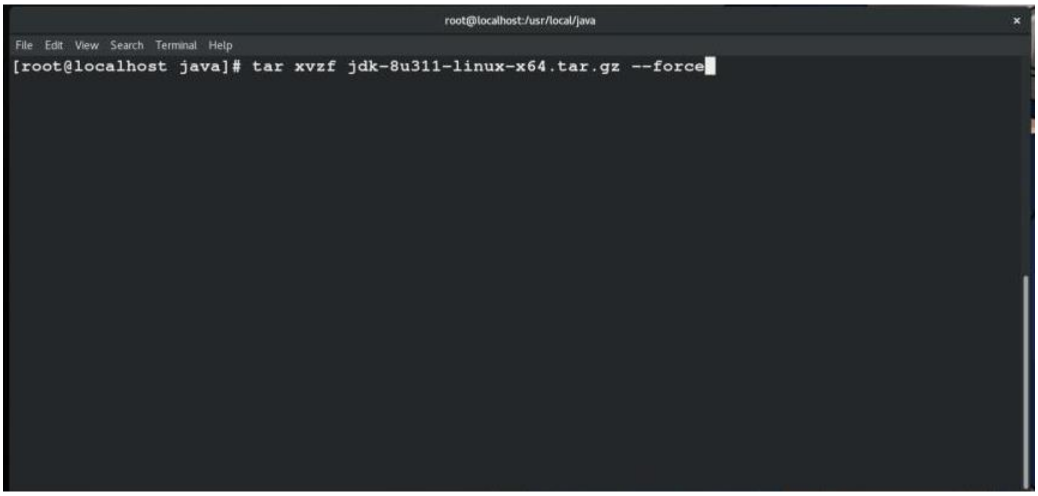


```
[root@localhost ~]# mkdir /usr/local/java
```



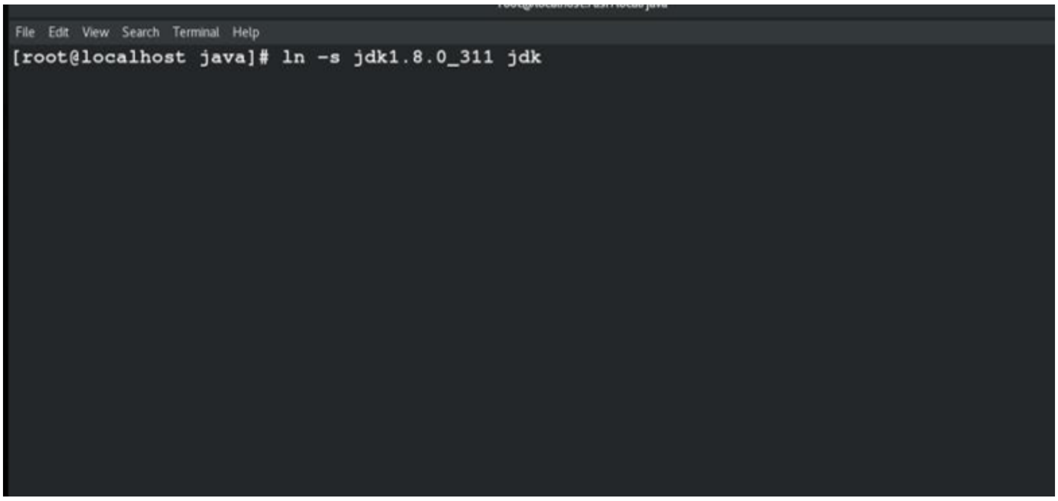
```
[root@localhost ~]# cp jdk-8u311-linux-x64.tar.gz /usr/local/java
```

Extract the jdk-8u311-linux-x64.tar.gz


root@localhost:/usr/local/java
File Edit View Search Terminal Help

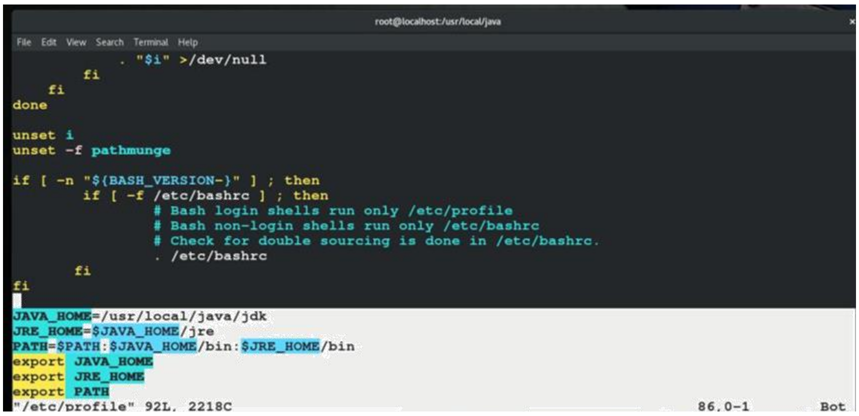
```
[root@localhost java]# tar xvzf jdk-8u311-linux-x64.tar.gz --force
```

Rename the jdk-8u311-linux-x64.tar.gz into jdk



```
[root@localhost java]# ln -s jdk1.8.0_311 jdk
```

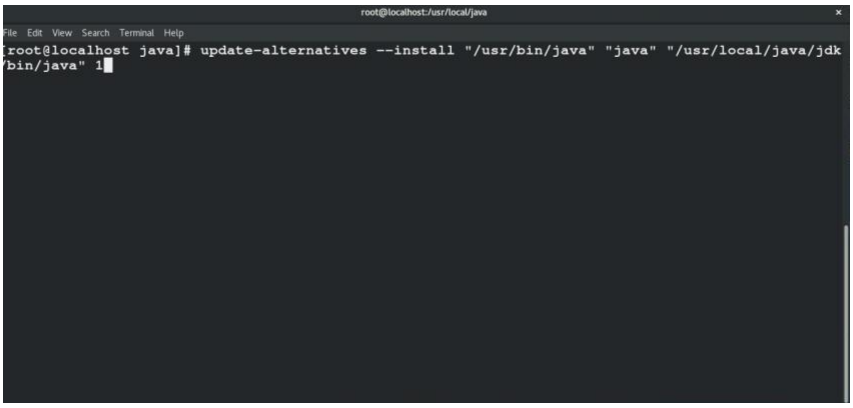
In the /etc/profiles we will also set up some of the required system variables and further inform our system regarding those updates. We also need to set oracle java as the default java



root@localhost:/usr/local/java
File Edit View Search Terminal Help

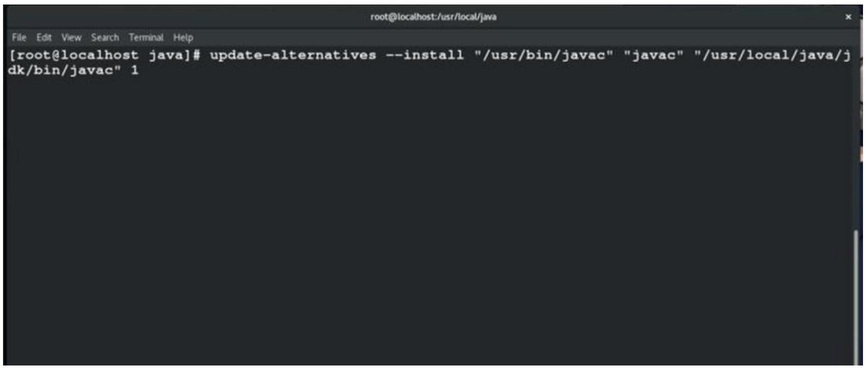
```
                . "$i" >/dev/null
            fi
        fi
done

unset i
unset -f pathmunge

if [ -n "${BASH_VERSION-}" ] ; then
        if [ -f /etc/bashrc ] ; then
                # Bash login shells run only /etc/profile
                # Bash non-login shells run only /etc/bashrc
                # Check for double sourcing is done in /etc/bashrc.
                . /etc/bashrc
        fi
fi

JAVA_HOME=/usr/local/java/jdk
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
"/etc/profile" 92L, 2218C                                         86,0-1        Bot
```

Now we need to update and install the alternatives variable for java and javac



root@localhost:/usr/local/java
File Edit View Search Terminal Help

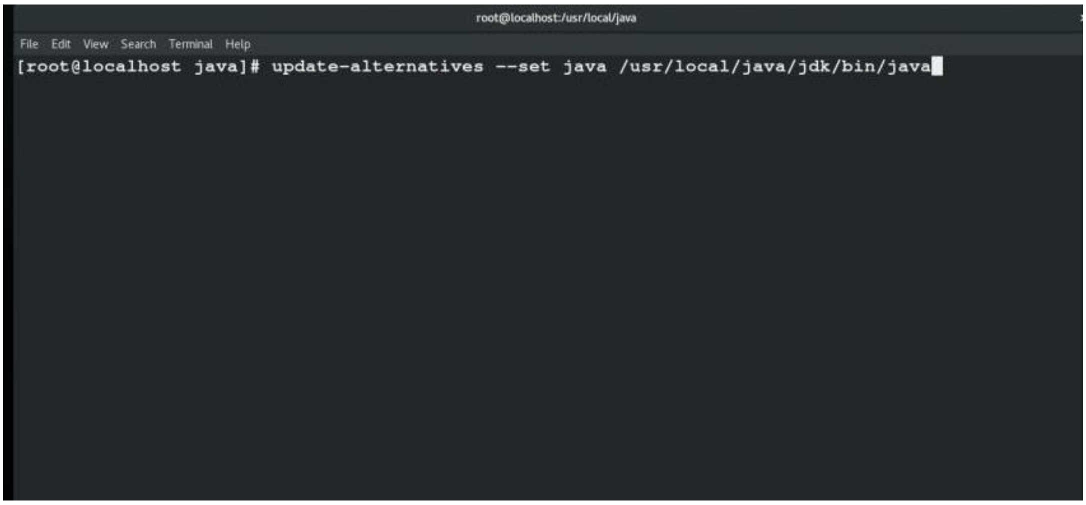
```
[root@localhost java]# update-alternatives --install "/usr/bin/java" "java" "/usr/local/java/jdk
/bin/java" 1
```



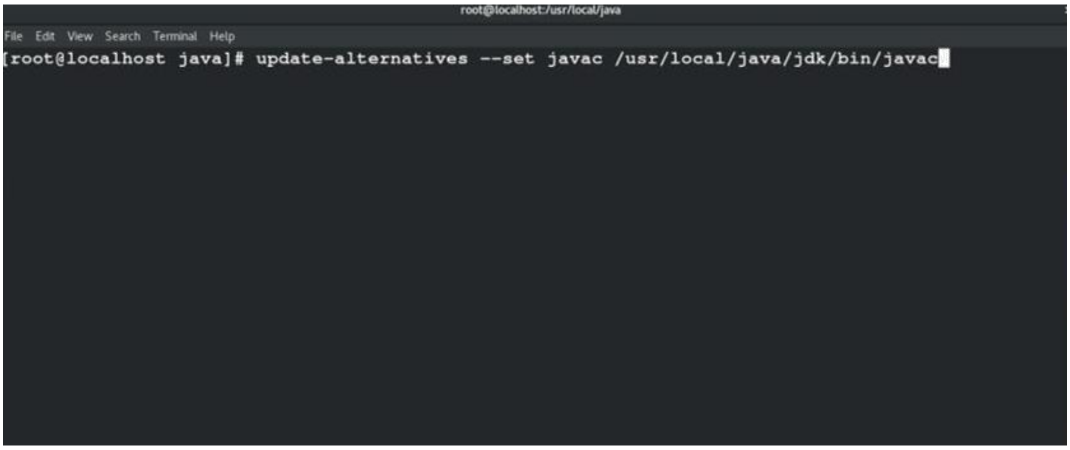
root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# update-alternatives --install "/usr/bin/javac" "javac" "/usr/local/java/j
dk/bin/javac" 1
```

Now we need to set the java and javac

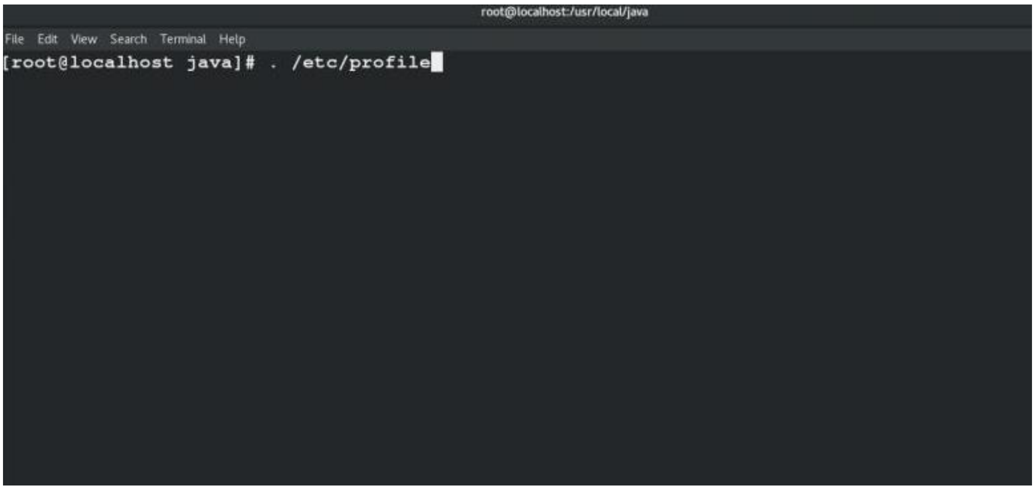

root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# update-alternatives --set java /usr/local/java/jdk/bin/java
```

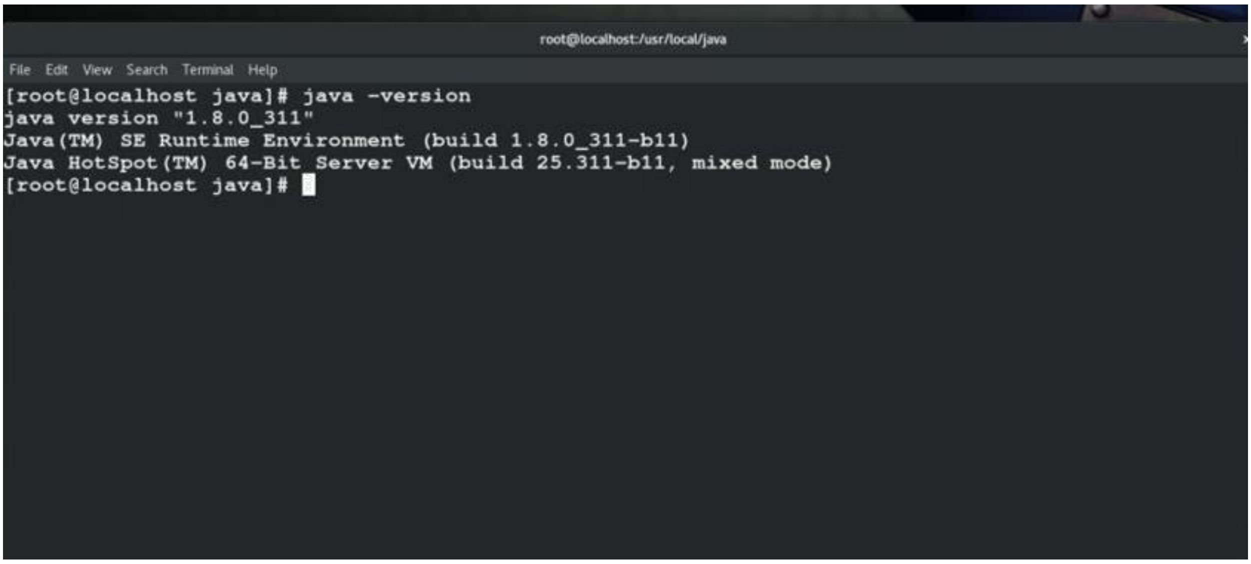

root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# update-alternatives --set javac /usr/local/java/jdk/bin/javac
```

Once the necessary things are done we will be restarting the /etc/profiles so that the updates will be implemented


root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# . /etc/profile
```



root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# java -version
java version "1.8.0_311"
Java(TM) SE Runtime Environment (build 1.8.0_311-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.311-b11, mixed mode)
[root@localhost java]# 
```

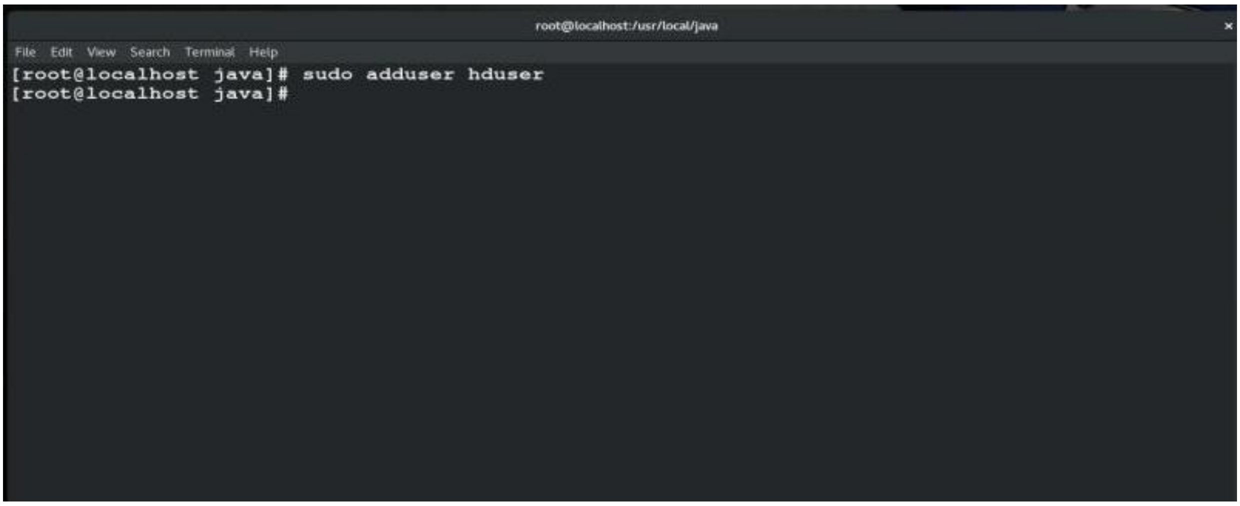
Java install successfully in our system

Set Up a Non-Root User for Hadoop Environment

It is advisable to create a non-root user, specifically for the Hadoop environment. A distinct user improves security and helps you manage your cluster more efficiently. To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a passwordless SSH connection with the localhost.

Create RedHat User

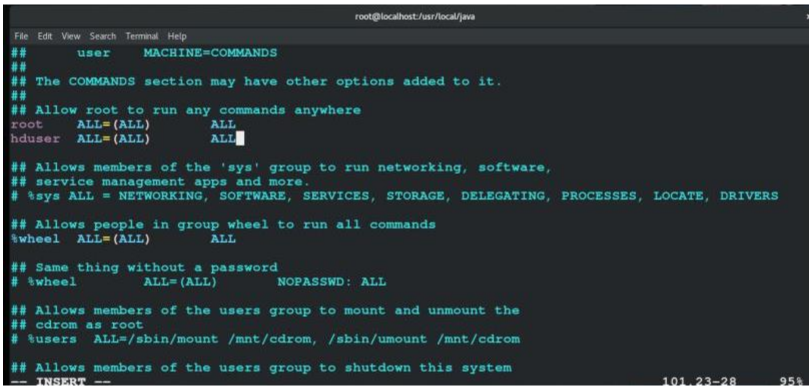
Utilize the **adduser** command to create a new Hadoop user:



root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# sudo adduser hduser
[root@localhost java]#
```

There are multiple situations where hduser might need the root power so for this we need to do the necessary updates in the /etc/sudoer file



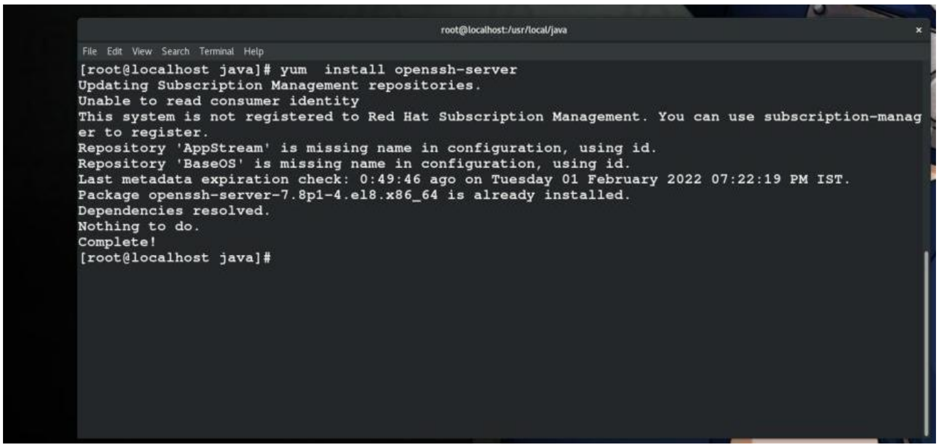
root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
hduser  ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
-- INSERT --                                                  101,23-28     95%
```

Install OpenSSH on Redhat

Install the OpenSSH server and client using the following command:

sudo yum install openssh-server openssh-client -y



root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# yum  install openssh-server
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manag
er to register.
Repository 'AppStream' is missing name in configuration, using id.
Repository 'BaseOS' is missing name in configuration, using id.
Last metadata expiration check: 0:49:46 ago on Tuesday 01 February 2022 07:22:19 PM IST.
Package openssh-server-7.8p1-4.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost java]#
```

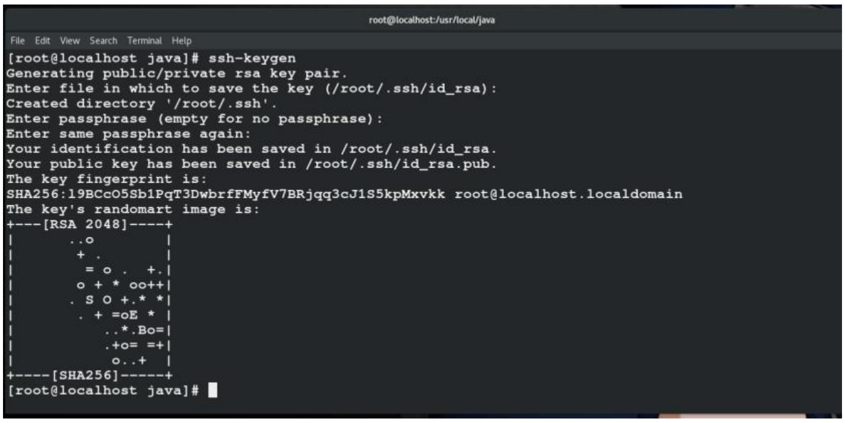
Enable Passwordless SSH for Hadoop User

Generate an SSH key pair and define the location is is to be stored in:

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

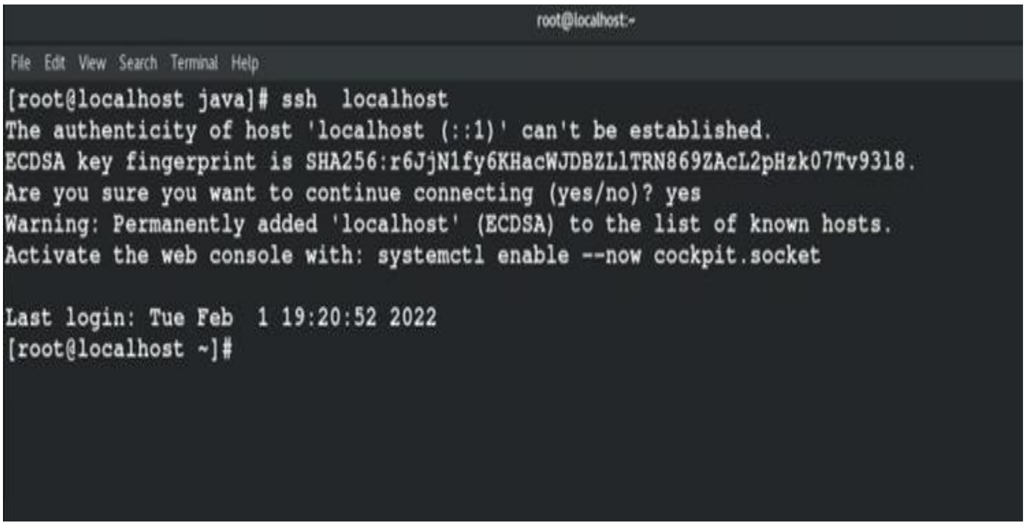
The system proceeds to generate and save the SSH key pair.

Use the cat command to store the public key as authorized_keys in the *ssh* directory:



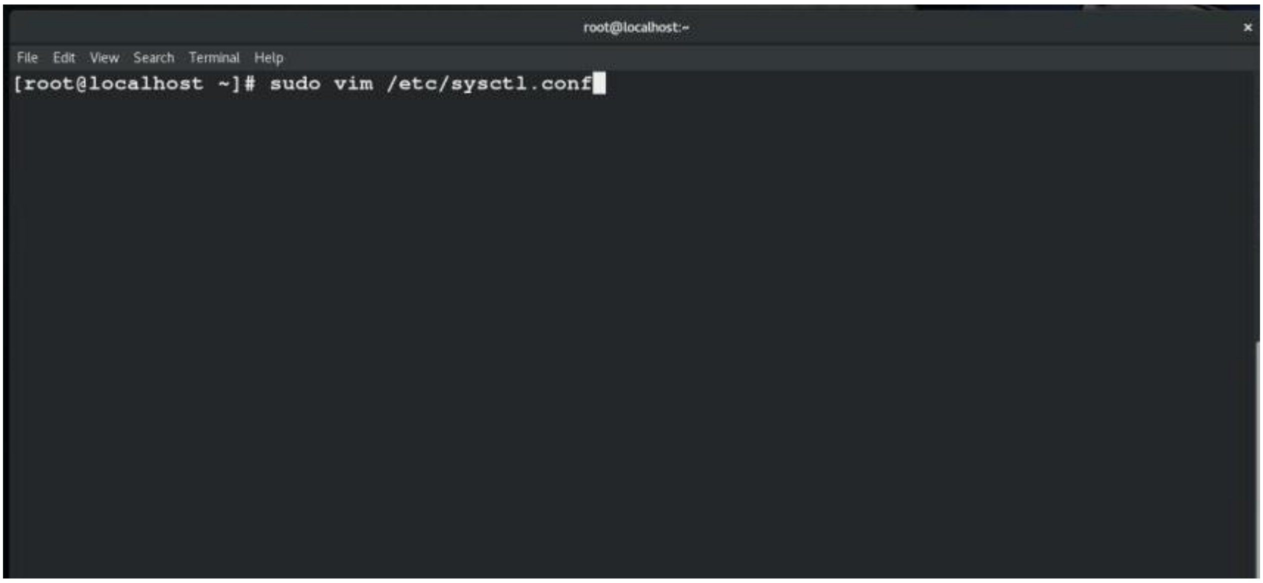
root@localhost:/usr/local/java
File Edit View Search Terminal Help

```
[root@localhost java]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:19BCcO5Sb1PqT3DwbrfFMyfV7BRjqq3cJ1S5kpMxvkk root@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]----+
|      ..o        |
|       + .       |
|        = o .  +.|
|       o + * oo++|
|      . S O +.* *|
|       . + =oE * |
|          ..*.Bo=|
|          .+o= =+|
|           o..+  |
+----[SHA256]-----+
[root@localhost java]#
```

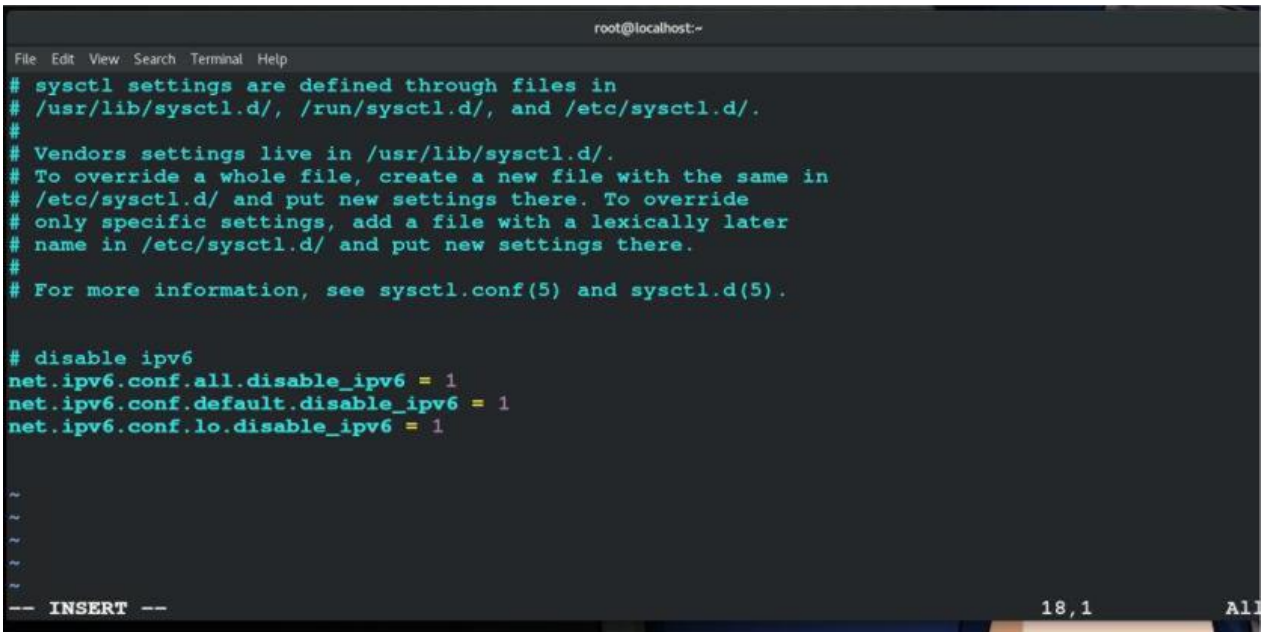


```
[root@localhost java]# ssh  localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:r6JjN1fy6KHacWJDBZLlTRN869ZAcL2pHzk07Tv9318.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Feb  1 19:20:52 2022
[root@localhost ~]#
```

We will also disable the ipv6 and only use the ipv4 in the machine

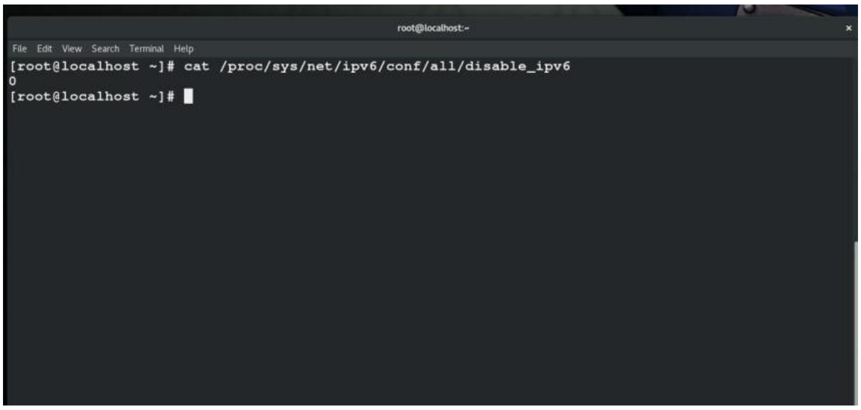


```
[root@localhost ~]# sudo vim /etc/sysctl.conf
```



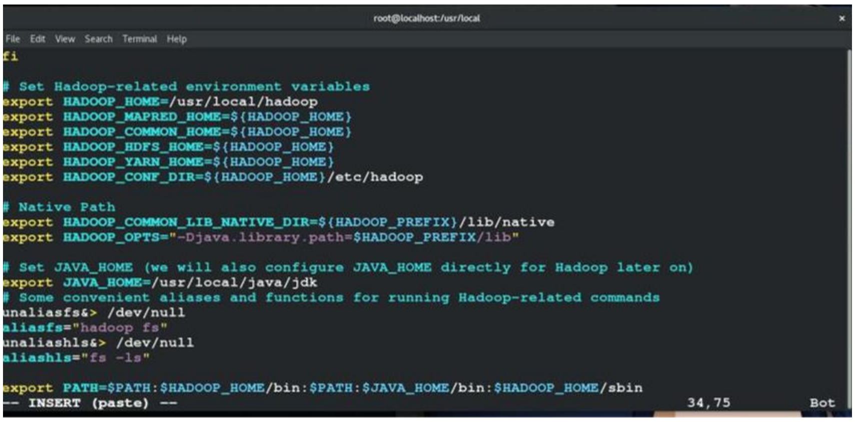
```
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).


# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1


~
~
~
~
~
-- INSERT --                                              18,1          All
```

to crosscheck we can use cat over /proc/sys/net/ipv6/conf/all/disable_ipv6 file

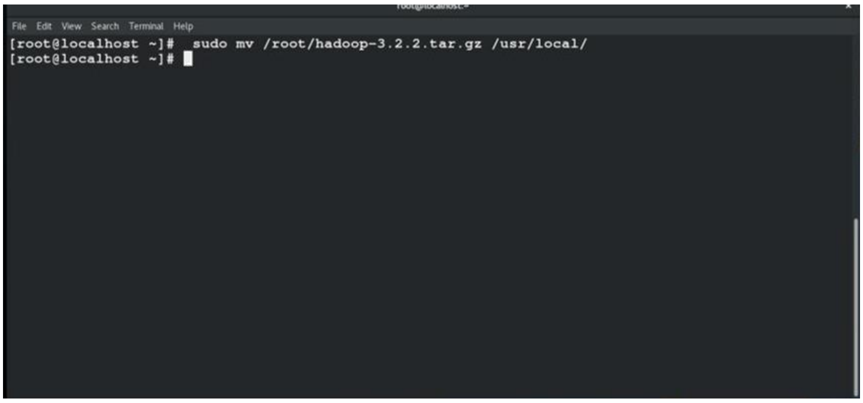


```
[root@localhost ~]# cat /proc/sys/net/ipv6/conf/all/disable_ipv6
0
[root@localhost ~]# 
```

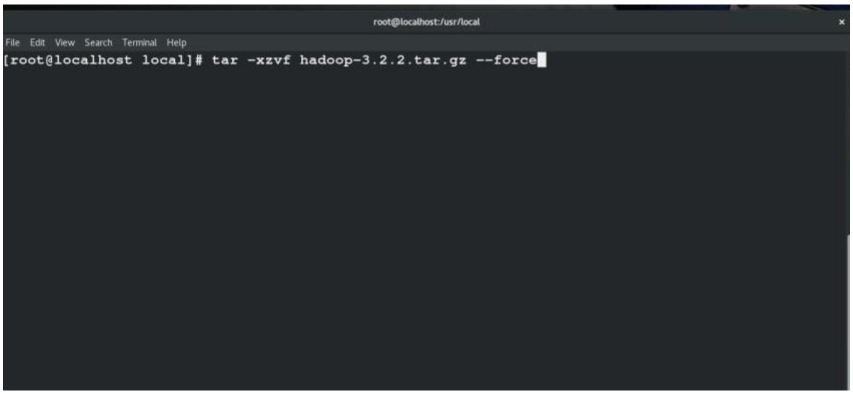
Move hadoop tar file from ~ file to /usr/local



```
fi

# Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export HADOOP_YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

# Native Path
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"

# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/usr/local/java/jdk
# Some convenient aliases and functions for running Hadoop-related commands
unaliasfs&> /dev/null
aliasfs="hadoop fs"
unaliashls&> /dev/null
aliashls="fs -ls"

export PATH=$PATH:$HADOOP_HOME/bin:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/sbin
-- INSERT (paste) --                                           34,75         Bot
```



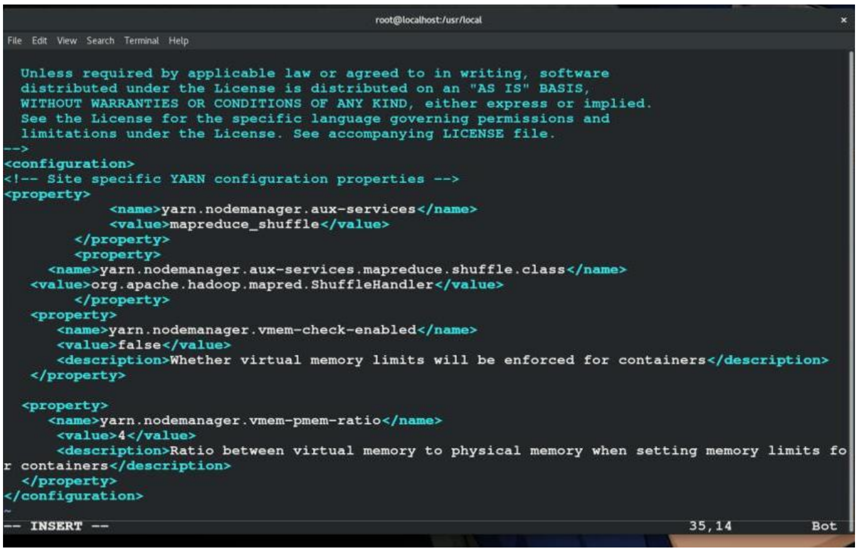
```
[root@localhost ~]#  sudo mv /root/hadoop-3.2.2.tar.gz /usr/local/
[root@localhost ~]# 
```



root@localhost:/usr/local
File Edit View Search Terminal Help

```
[root@localhost local]# tar -xzvf hadoop-3.2.2.tar.gz --force
```

Now its time for the hadoop files
Configuration Changes in yarn-site.xml file
Edit **yarn-site.xml** with the following entries.



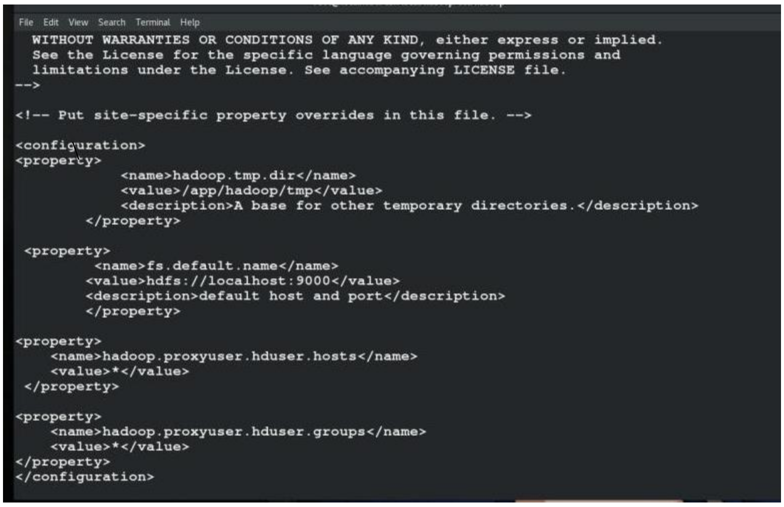
root@localhost:/usr/local
File Edit View Search Terminal Help

```
  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.aux-services.mapreduce.shuffle.class</name>
   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
        </property>
   <property>
      <name>yarn.nodemanager.vmem-check-enabled</name>
      <value>false</value>
      <description>Whether virtual memory limits will be enforced for containers</description>
   </property>

  <property>
     <name>yarn.nodemanager.vmem-pmem-ratio</name>
      <value>4</value>
      <description>Ratio between virtual memory to physical memory when setting memory limits fo
r containers</description>
  </property>
</configuration>
~
-- INSERT --                                                          35,14         Bot
```

Configuration Changes in core-site.xml file

Edit the **core-site.xml** with vim or you can use any of the editors. The file is under

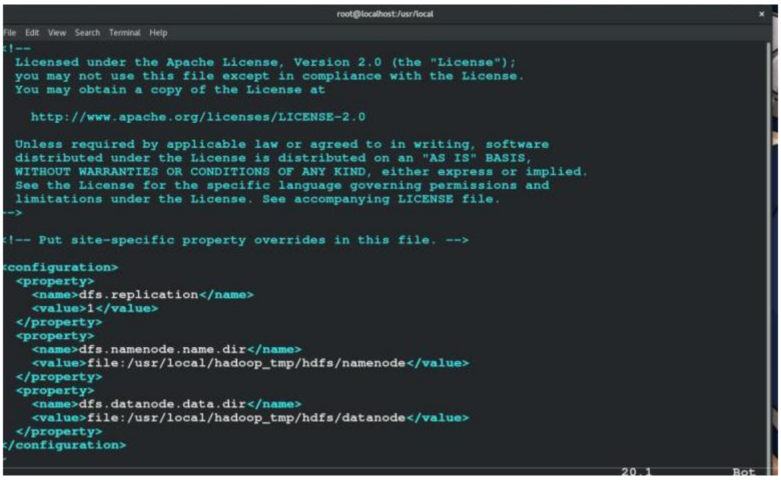
/etc/hadoop inside **hadoop** home directory and add following entries.



```
  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>hadoop.tmp.dir</name>
            <value>/app/hadoop/tmp</value>
            <description>A base for other temporary directories.</description>
        </property>

 <property>
         <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
        <description>default host and port</description>
        </property>

<property>
    <name>hadoop.proxyuser.hduser.hosts</name>
    <value>*</value>
 </property>

<property>
    <name>hadoop.proxyuser.hduser.groups</name>
    <value>*</value>
</property>
</configuration>
```

Configuration Changes in mapred-site.xml file

Copy the mapred-site.xml from mapred-site.xml.template using cp command and then edit the mapred-site.xml placed in /etc/hadoop under hadoop installation directory with the following changes.



root@localhost:/usr/local
File Edit View Search Terminal Help

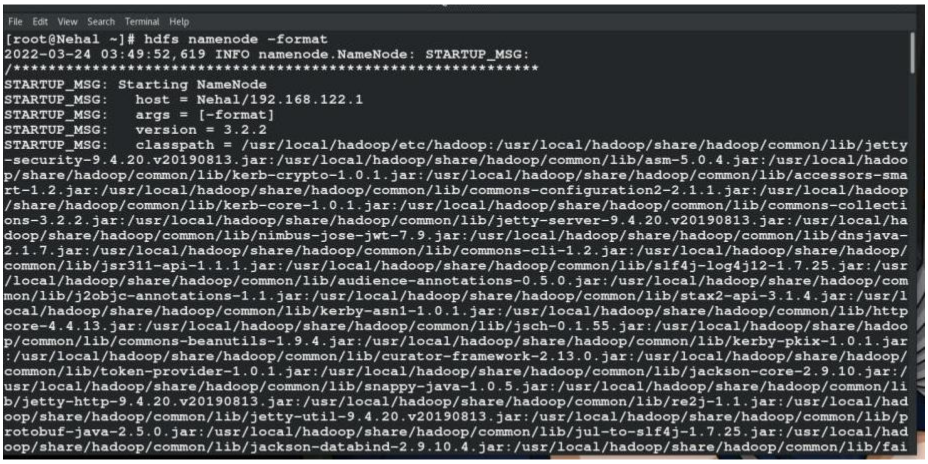
```
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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.
-->

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

<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>
                                                              20,1          Bot
```

Now create a namenode and datanode folder and provide the all the necessary permission to it

```
# Sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
# Sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

Starting the Hadoop Cluster

Format the namenode before using it for the first time. As hadoop users run the below command to format the Namenode.



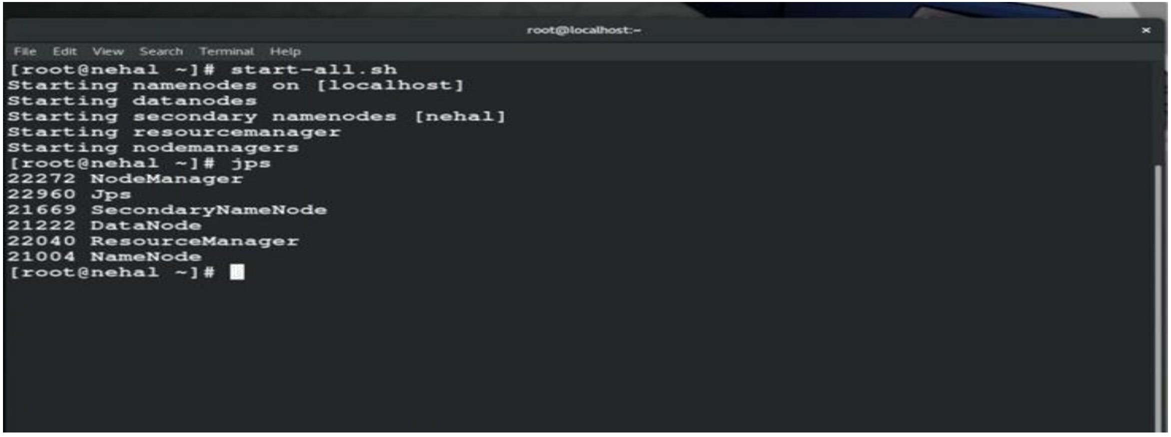
```
[root@Nehal ~]# hdfs namenode -format
2022-03-24 03:49:52,619 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Nehal/192.168.122.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.2
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/jetty
-security-9.4.20.v20190813.jar:/usr/local/hadoop/share/hadoop/common/lib/asm-5.0.4.jar:/usr/local/hadoo
p/share/hadoop/common/lib/kerb-crypto-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/accessors-sma
rt-1.2.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-configuration2-2.1.1.jar:/usr/local/hadoop
/share/hadoop/common/lib/kerb-core-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-collecti
ons-3.2.2.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-server-9.4.20.v20190813.jar:/usr/local/ha
doop/share/hadoop/common/lib/nimbus-jose-jwt-7.9.jar:/usr/local/hadoop/share/hadoop/common/lib/dnsjava-
2.1.7.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/
common/lib/jsr311-api-1.1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar:/usr
/local/hadoop/share/hadoop/common/lib/audience-annotations-0.5.0.jar:/usr/local/hadoop/share/hadoop/com
mon/lib/j2objc-annotations-1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/stax2-api-3.1.4.jar:/usr/l
ocal/hadoop/share/hadoop/common/lib/kerby-asn1-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/http
core-4.4.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jsch-0.1.55.jar:/usr/local/hadoop/share/hadoo
p/common/lib/commons-beanutils-1.9.4.jar:/usr/local/hadoop/share/hadoop/common/lib/kerby-pkix-1.0.1.jar
:/usr/local/hadoop/share/hadoop/common/lib/curator-framework-2.13.0.jar:/usr/local/hadoop/share/hadoop/
common/lib/token-provider-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-core-2.9.10.jar:/
usr/local/hadoop/share/hadoop/common/lib/snappy-java-1.0.5.jar:/usr/local/hadoop/share/hadoop/common/li
b/jetty-http-9.4.20.v20190813.jar:/usr/local/hadoop/share/hadoop/common/lib/re2j-1.1.jar:/usr/local/had
oop/share/hadoop/common/lib/jetty-util-9.4.20.v20190813.jar:/usr/local/hadoop/share/hadoop/common/lib/p
rotobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/common/lib/jul-to-slf4j-1.7.25.jar:/usr/local/had
oop/share/hadoop/common/lib/jackson-databind-2.9.10.4.jar:/usr/local/hadoop/share/hadoop/common/lib/fai
```

Once the Namenode has been formatted then start the HDFS using the

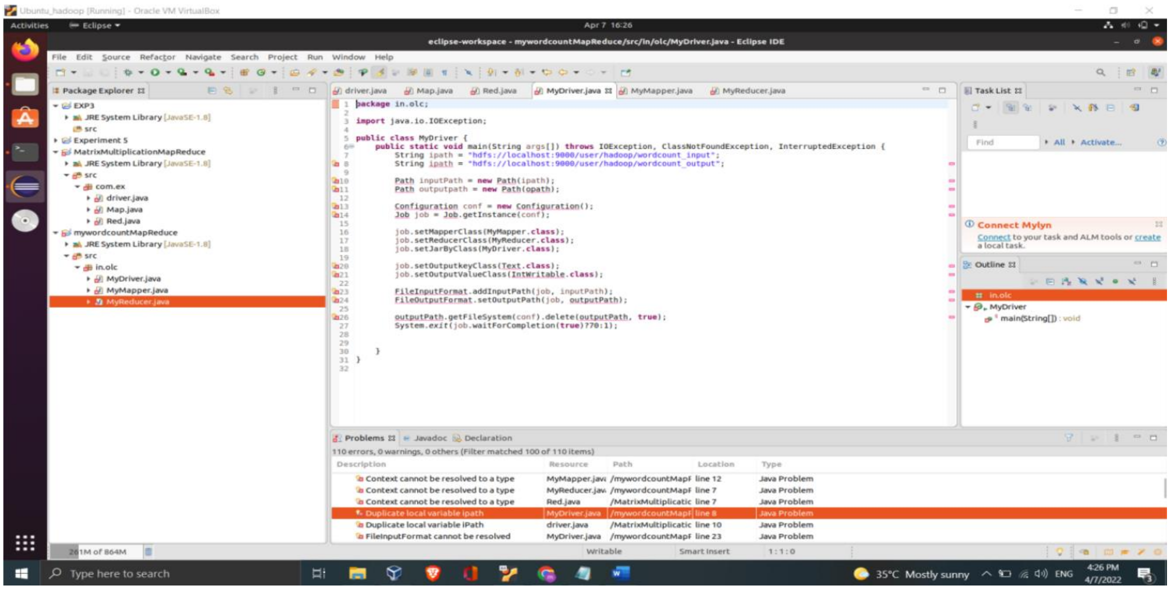
```
$ start-all.sh
```

All Services started successfully and all the node are

Working



```
[root@nehal ~]# start-all.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [nehal]
Starting resourcemanager
Starting nodemanagers
[root@nehal ~]# jps
22272 NodeManager
22960 Jps
21669 SecondaryNameNode
21222 DataNode
22040 ResourceManager
21004 NameNode
[root@nehal ~]#
```

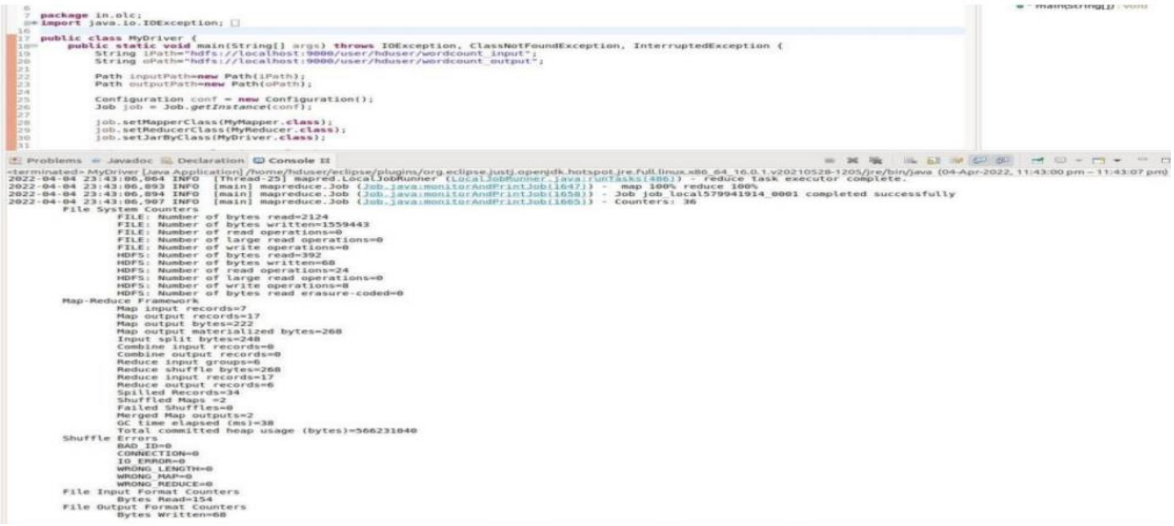


```java
package in.olc;

import java.io.IOException;

public class MyDriver {
    public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException {
        String ipath = "hdfs://localhost:9000/user/hadoop/wordcount_input";
        String ipath = "hdfs://localhost:9000/user/hadoop/wordcount_output";

        Path inputPath = new Path(ipath);
        Path outputpath = new Path(opath);

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf);

        job.setMapperClass(MyMapper.class);
        job.setReducerClass(MyReducer.class);
        job.setJarByClass(MyDriver.class);

        job.setOutputkeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, inputPath);
        FileOutputFormat.setOutputPath(job, outputPath);

        outputPath.getFileSystem(conf).delete(outputPath, true);
        System.exit(job.waitForCompletion(true)?70:1);


    }
}
```

Problems | Javadoc | Declaration

110 errors, 0 warnings, 0 others (Filter matched 100 of 110 items)

Description	Resource	Path	Location	Type
Context cannot be resolved to a type	MyMapper.jav	/mywordcountMapF	line 12	Java Problem
Context cannot be resolved to a type	MyReducer.jav	/mywordcountMapF	line 7	Java Problem
Context cannot be resolved to a type	Red.java	/MatrixMultiplicatic	line 7	Java Problem
Duplicate local variable ipath	MyDriver.java	/mywordcountMapF	line 8	Java Problem
Duplicate local variable iPath	driver.java	/MatrixMultiplicatic	line 10	Java Problem
FileInputFormat cannot be resolved	MyDriver.java	/mywordcountMapF	line 23	Java Problem



```java
package in.olc;
import java.io.IOException;

public class MyDriver {
    public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {
        String iPath="hdfs://localhost:9000/user/hduser/wordcount_input";
        String oPath="hdfs://localhost:9000/user/hduser/wordcount_output";

        Path inputPath=new Path(iPath);
        Path outputPath=new Path(oPath);

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf);

        job.setMapperClass(MyMapper.class);
        job.setReducerClass(MyReducer.class);
        job.setJarByClass(MyDriver.class);
```

```
<terminated> MyDriver [Java Application] /home/hduser/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (04-Apr-2022, 11:43:00 pm – 11:43:07 pm)
2022-04-04 23:43:06,864 INFO  [Thread-25] mapred.LocalJobRunner (LocalJobRunner.java:runTasks(486)) - reduce task executor complete.
2022-04-04 23:43:06,893 INFO  [main] mapreduce.Job (Job.java:monitorAndPrintJob(1647)) -  map 100% reduce 100%
2022-04-04 23:43:06,894 INFO  [main] mapreduce.Job (Job.java:monitorAndPrintJob(1658)) - Job job_local579941914_0001 completed successfully
2022-04-04 23:43:06,907 INFO  [main] mapreduce.Job (Job.java:monitorAndPrintJob(1665)) - Counters: 36
	File System Counters
		FILE: Number of bytes read=2124
		FILE: Number of bytes written=1559443
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=392
		HDFS: Number of bytes written=68
		HDFS: Number of read operations=24
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=8
		HDFS: Number of bytes read erasure-coded=0
	Map-Reduce Framework
		Map input records=7
		Map output records=17
		Map output bytes=222
		Map output materialized bytes=268
		Input split bytes=248
		Combine input records=0
		Combine output records=0
		Reduce input groups=6
		Reduce shuffle bytes=268
		Reduce input records=17
		Reduce output records=6
		Spilled Records=34
		Shuffled Maps =2
		Failed Shuffles=0
		Merged Map outputs=2
		GC time elapsed (ms)=38
		Total committed heap usage (bytes)=566231040
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=154
	File Output Format Counters
		Bytes Written=68
```

Conclusion: Thus we have installed Hadoop and implemented program using MapReduce.