**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,

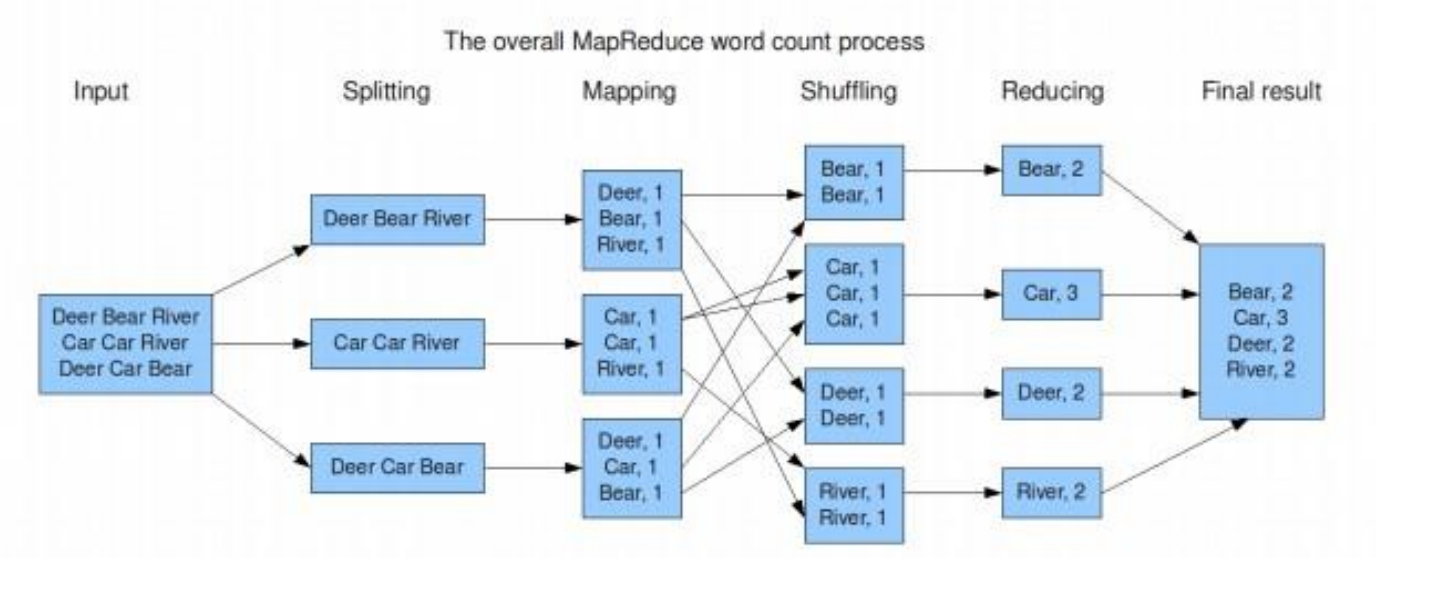
There key features and how the key-value pairs are generated in the Mapper.

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

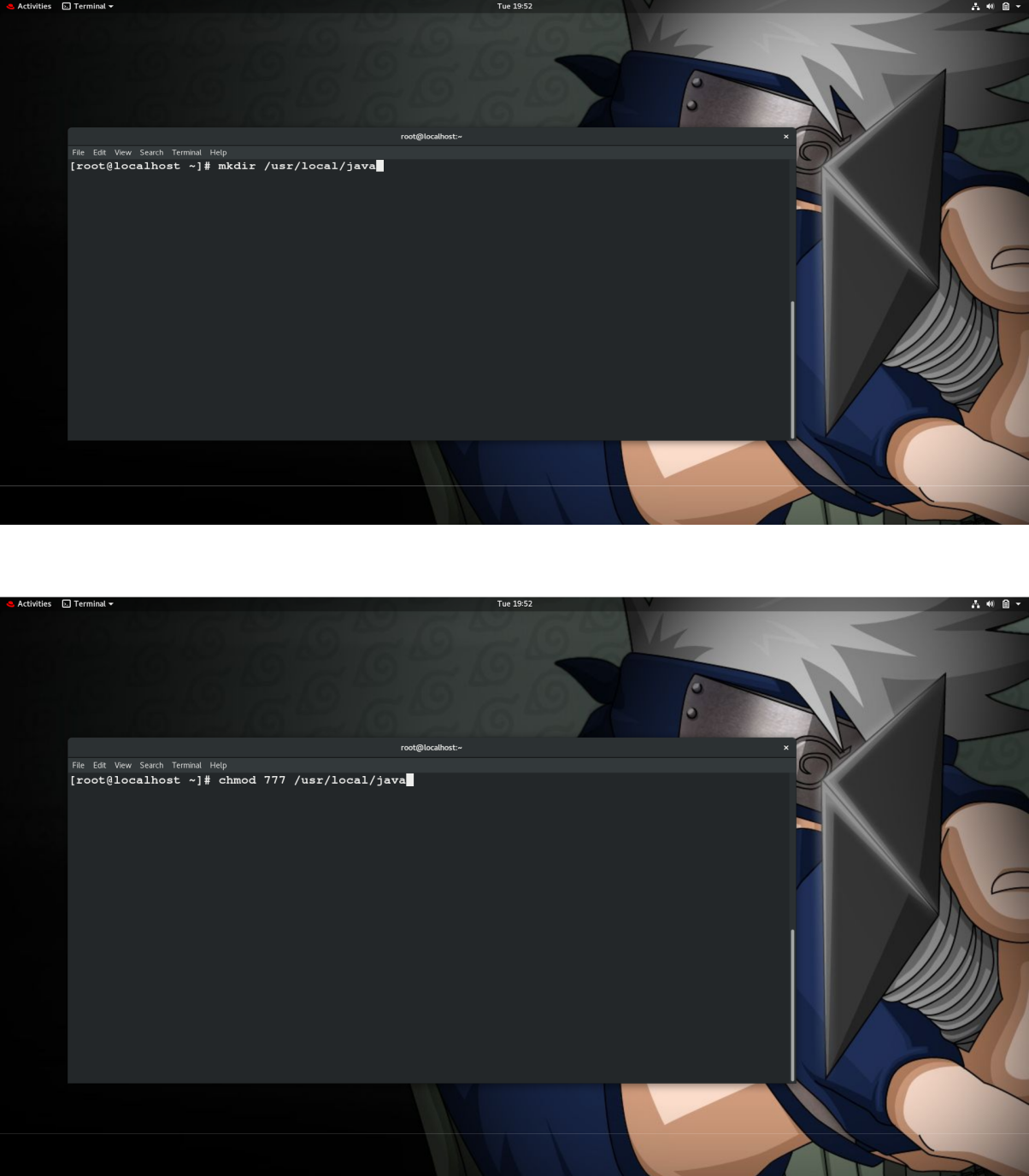
the keys of common values.

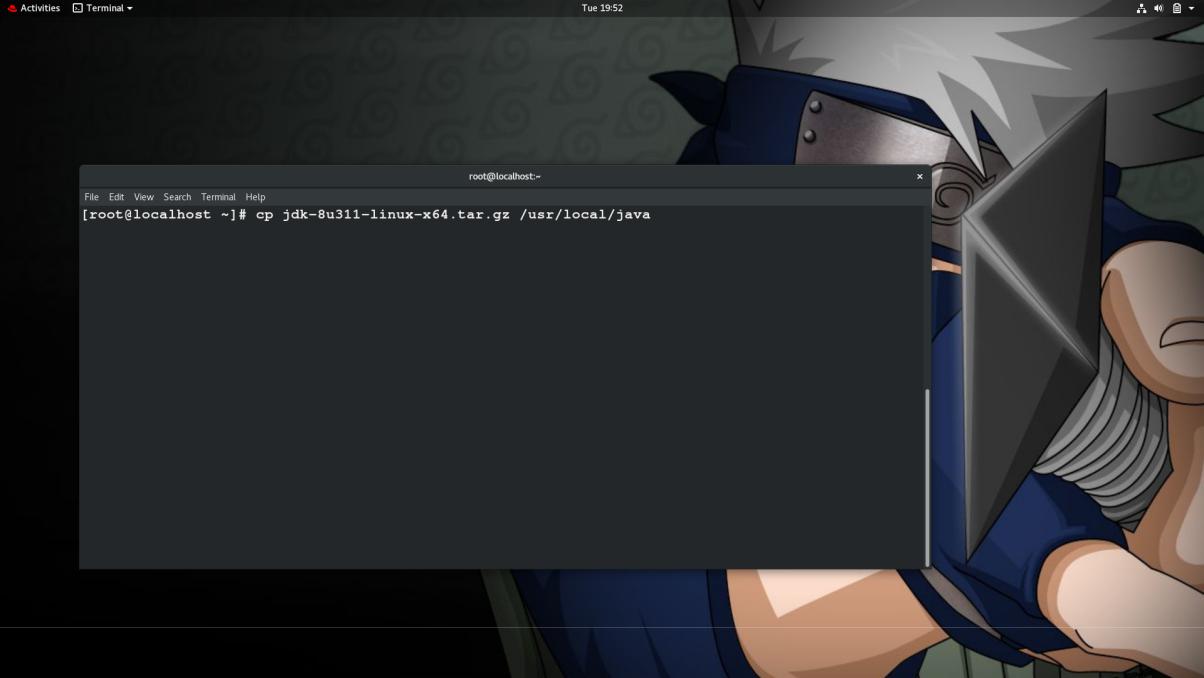
Example :-



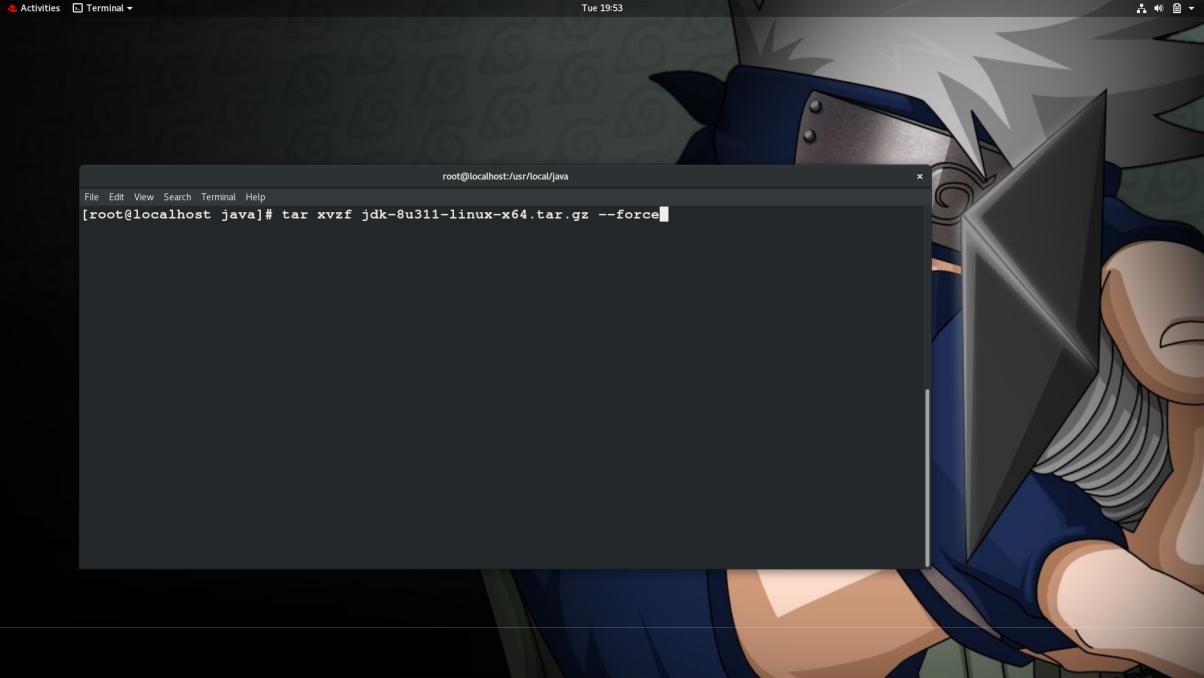
Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

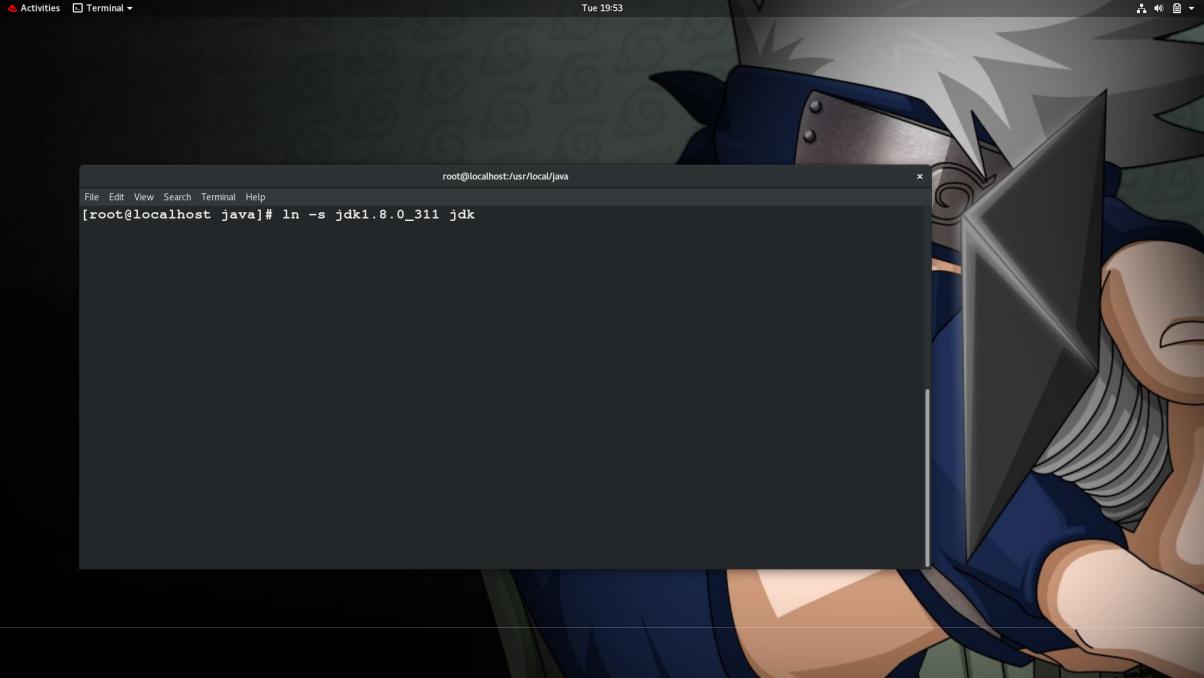
At the moment, Apache Hadoop 3.x fully supports Java 8. The OpenJDK 8 package in RedHat 8 contains both the runtime environment and development kit. Move jdk from desktop to /usr/local/java

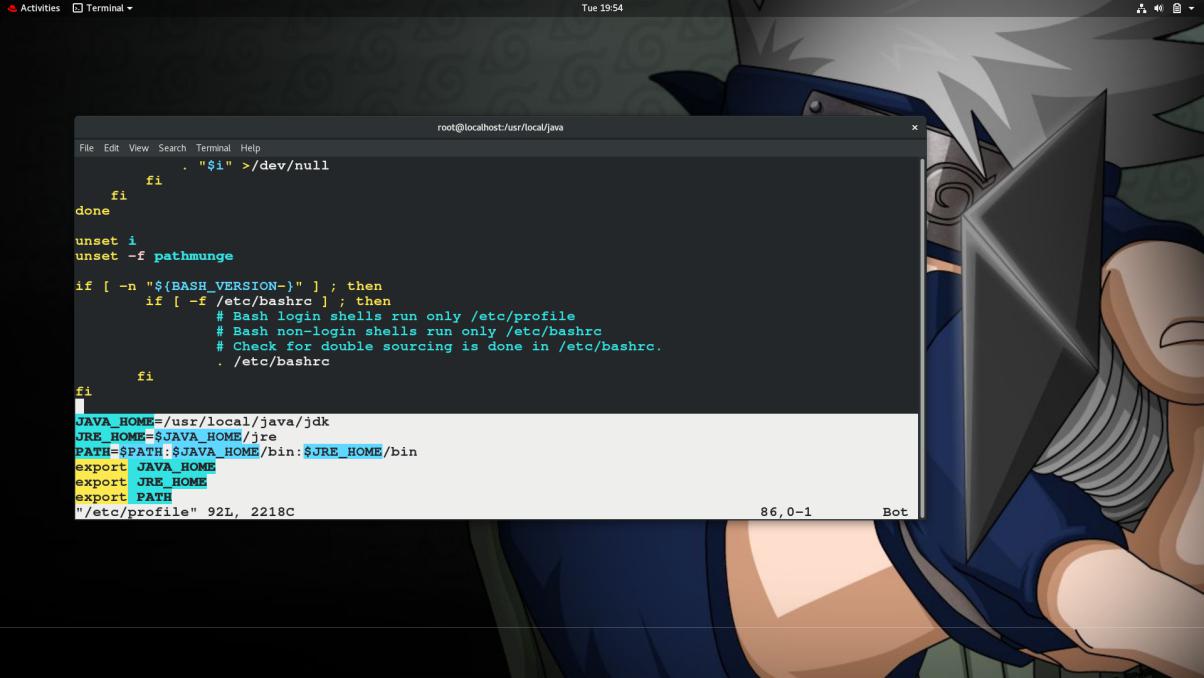
**Practical:** 



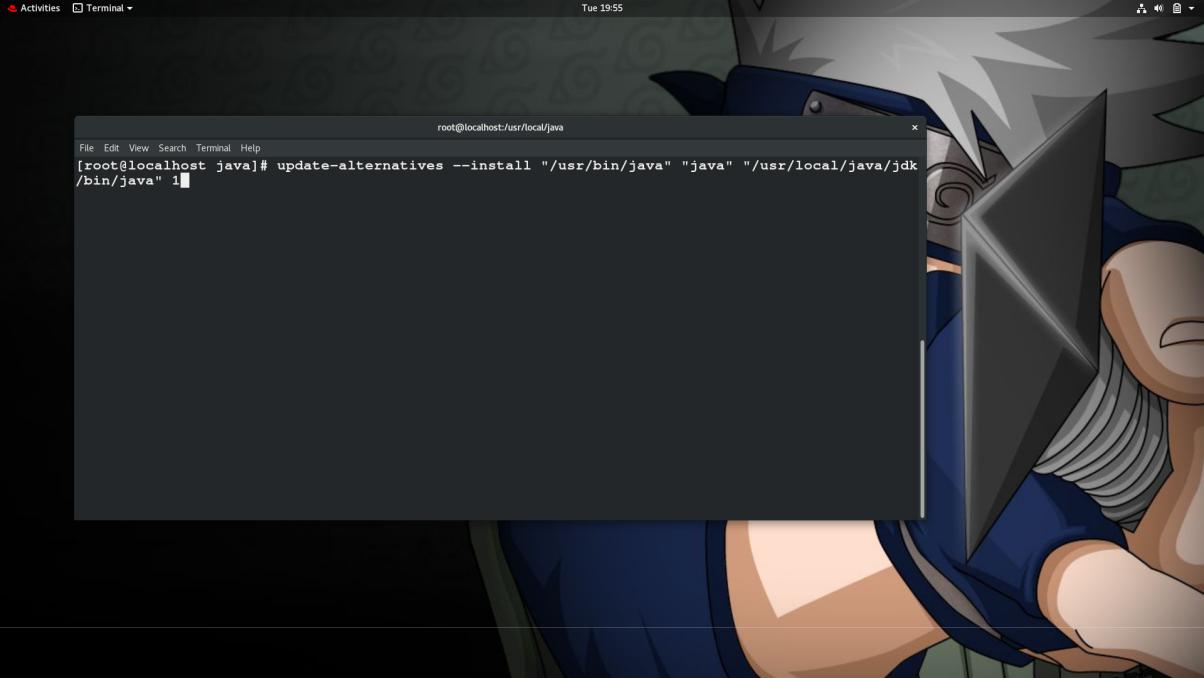
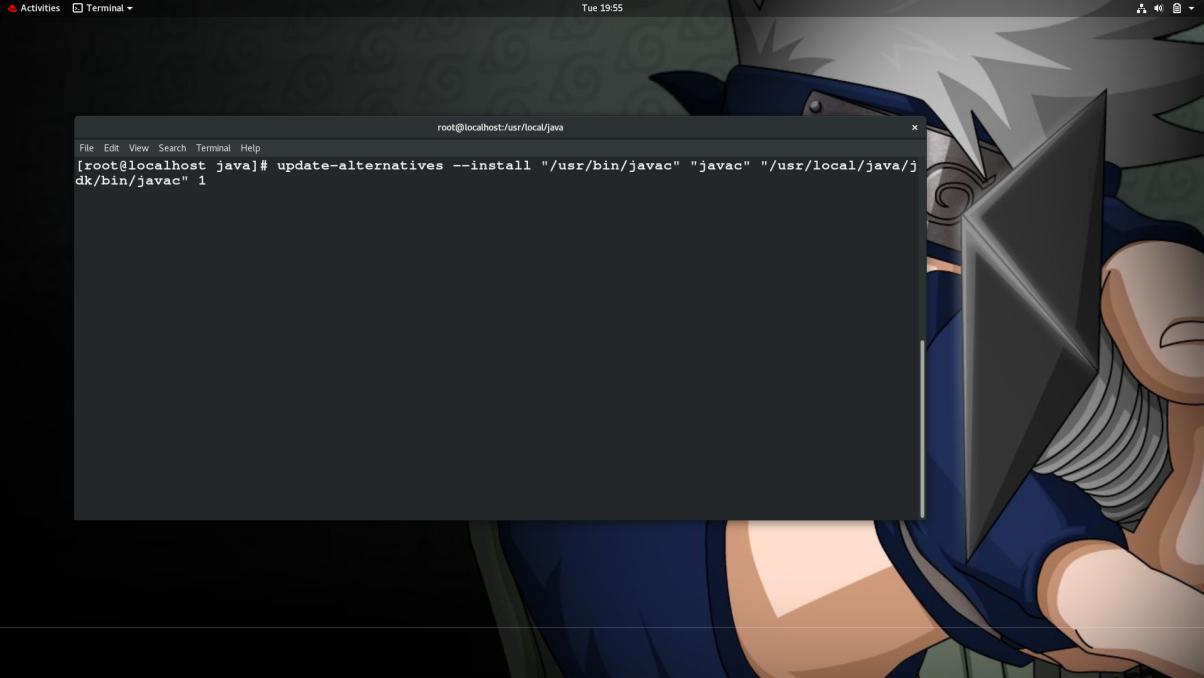
Extract the jdk-8u311-linux-x64.tar.gz



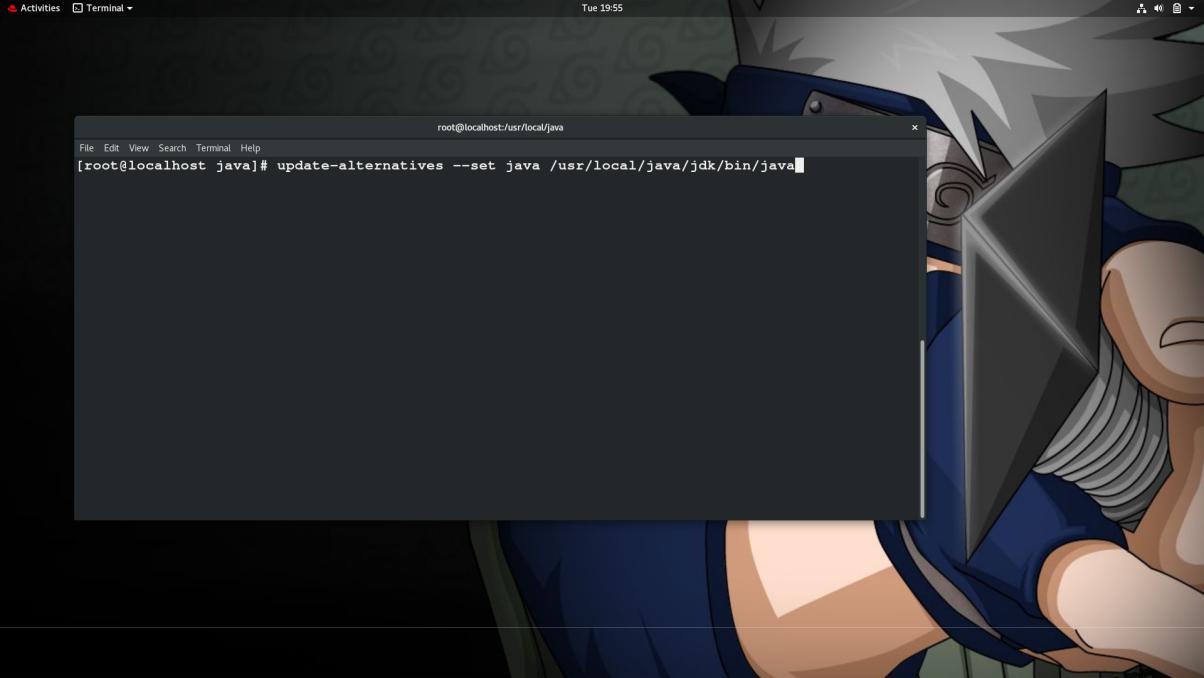
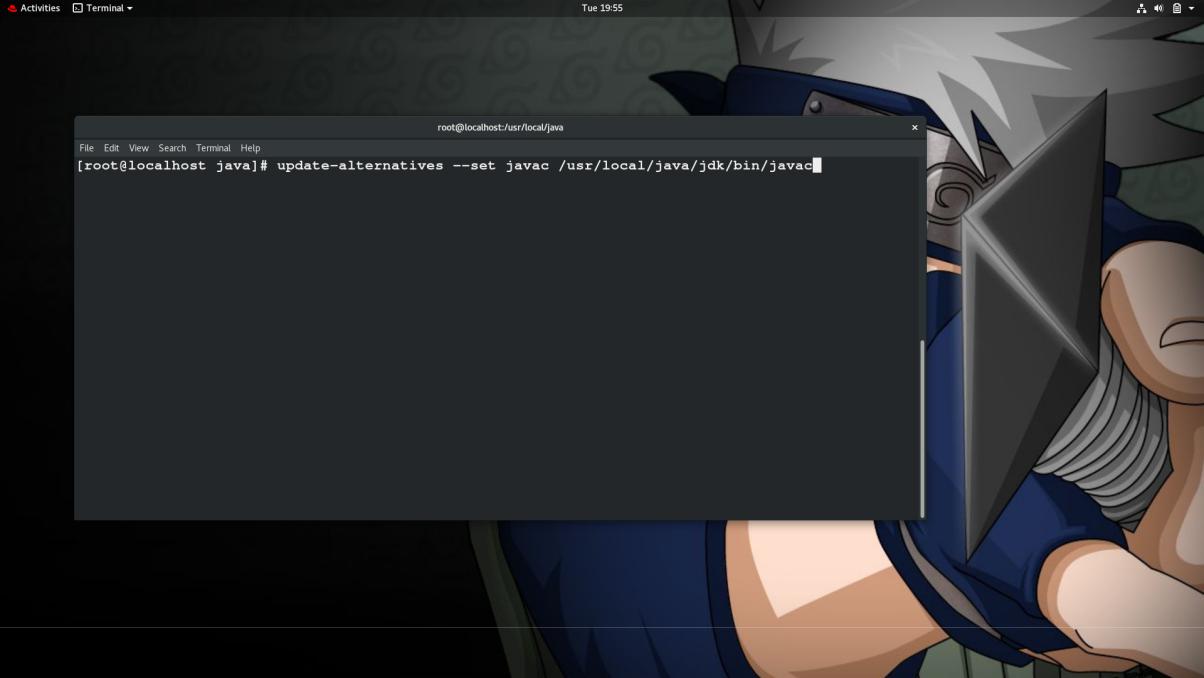
Rename the jdk-8u311-linux-x64.tar.gz into jdk

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

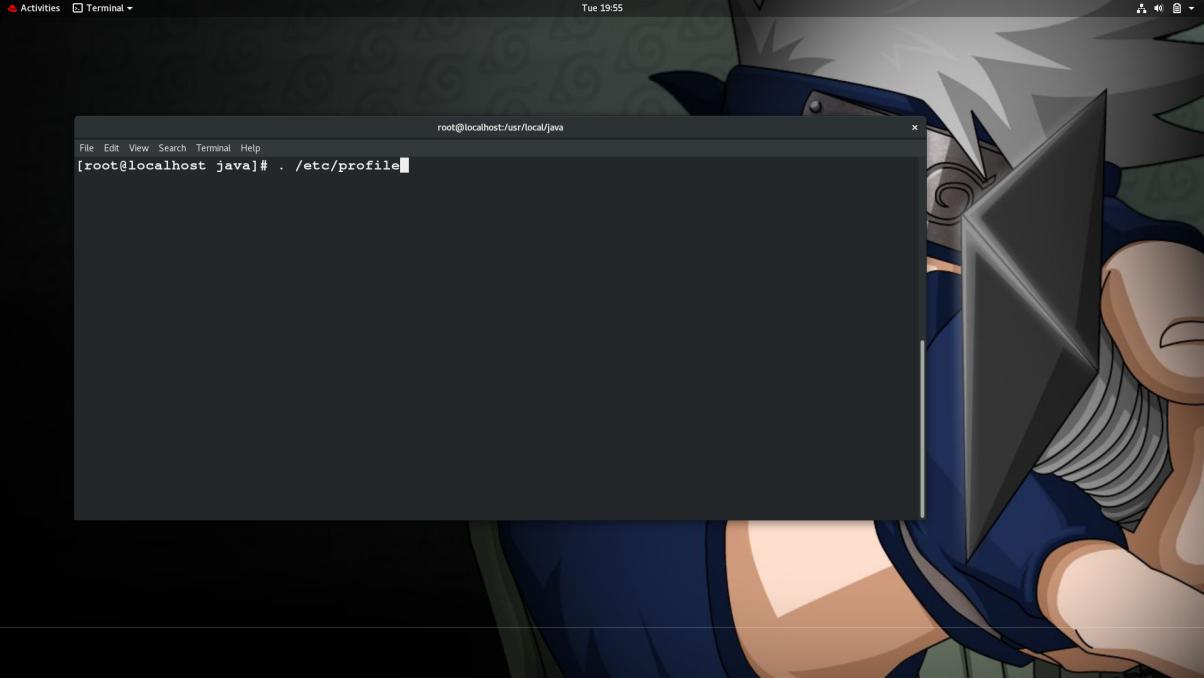
Now we need to update and install the alternatives variable for java and javac

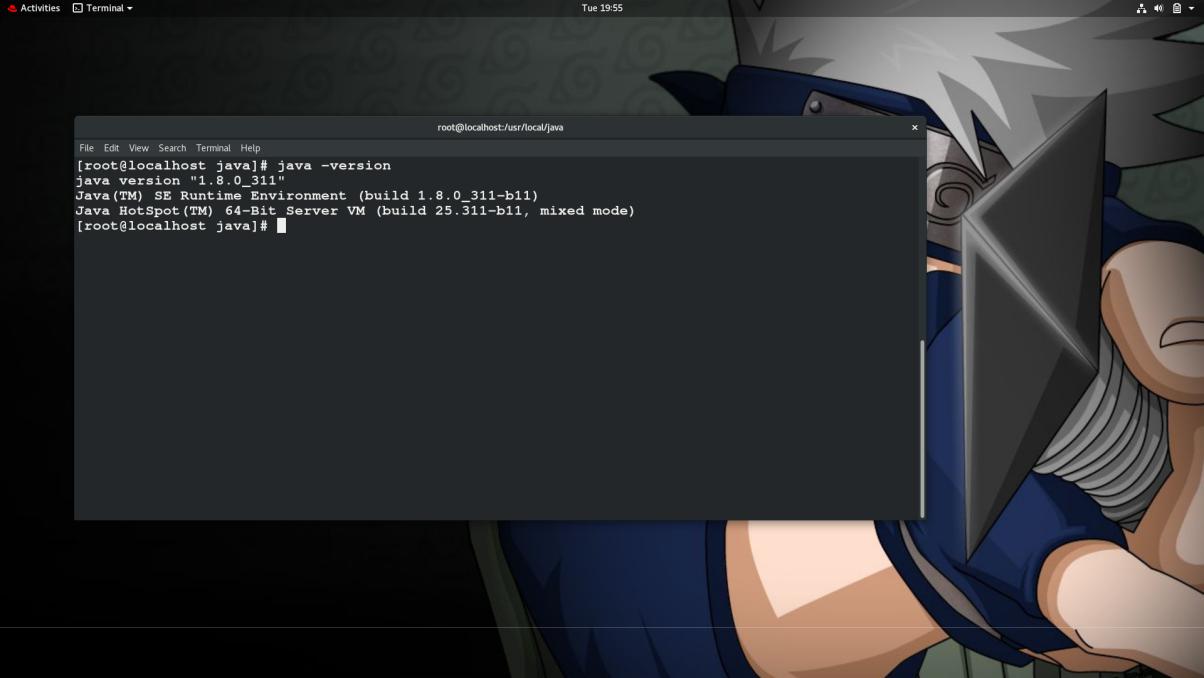


Now we need to set the java and javac



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





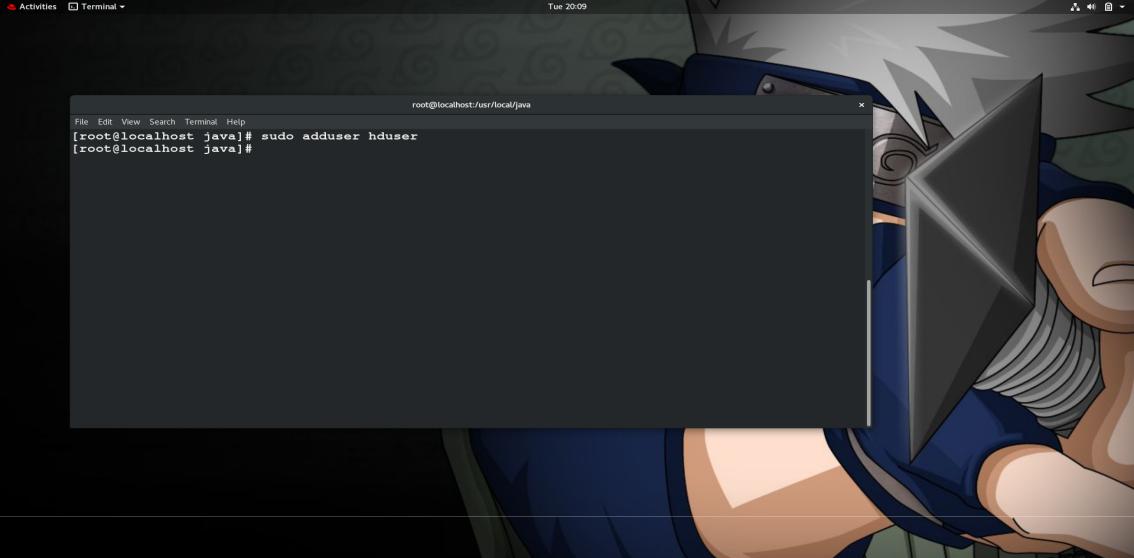
Java install successfully in our system

**Set Up a Non-Root User for Hadoop Environment**

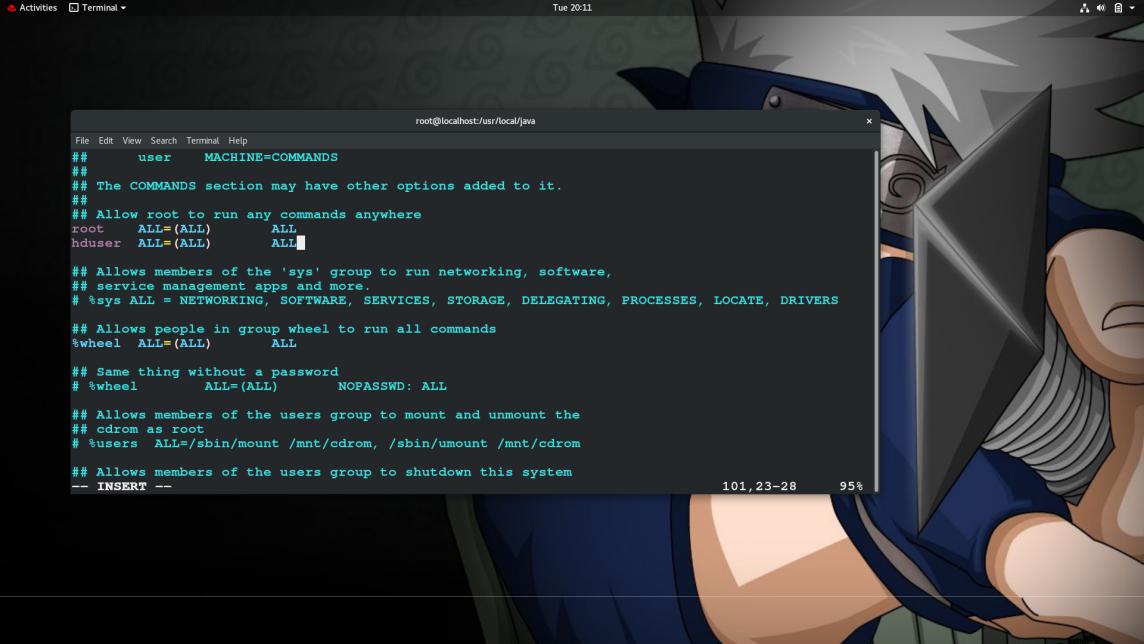
It is advisable to create a non-root user, specifically for the Hadoop environment. A distinct user improves security and helps you manage your cluster more efficiently. To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a passwordless SSH connection with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

**Create RedHat User**

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



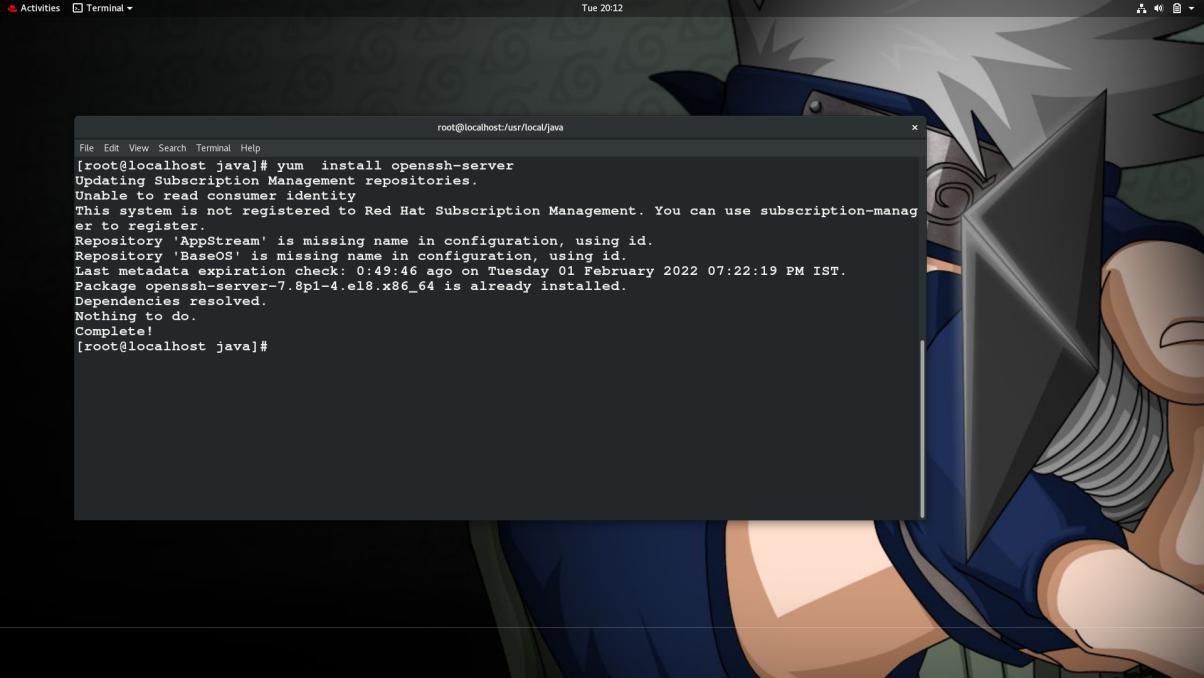
There are multiple situations where hduser might need the root power so for this we need to do the necessary updates in the /etc/sudoer file



**Install OpenSSH on Redhat**

Install the OpenSSH server and client using the following command:

**sudo yum install openssh-server openssh-client -y**



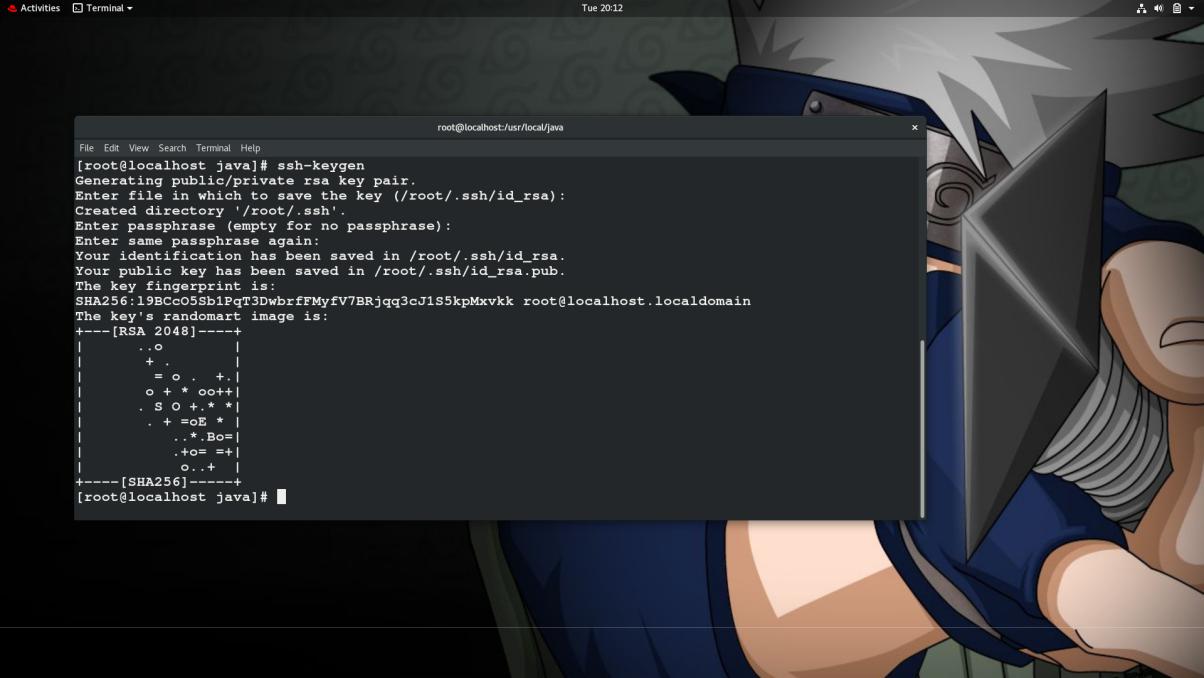
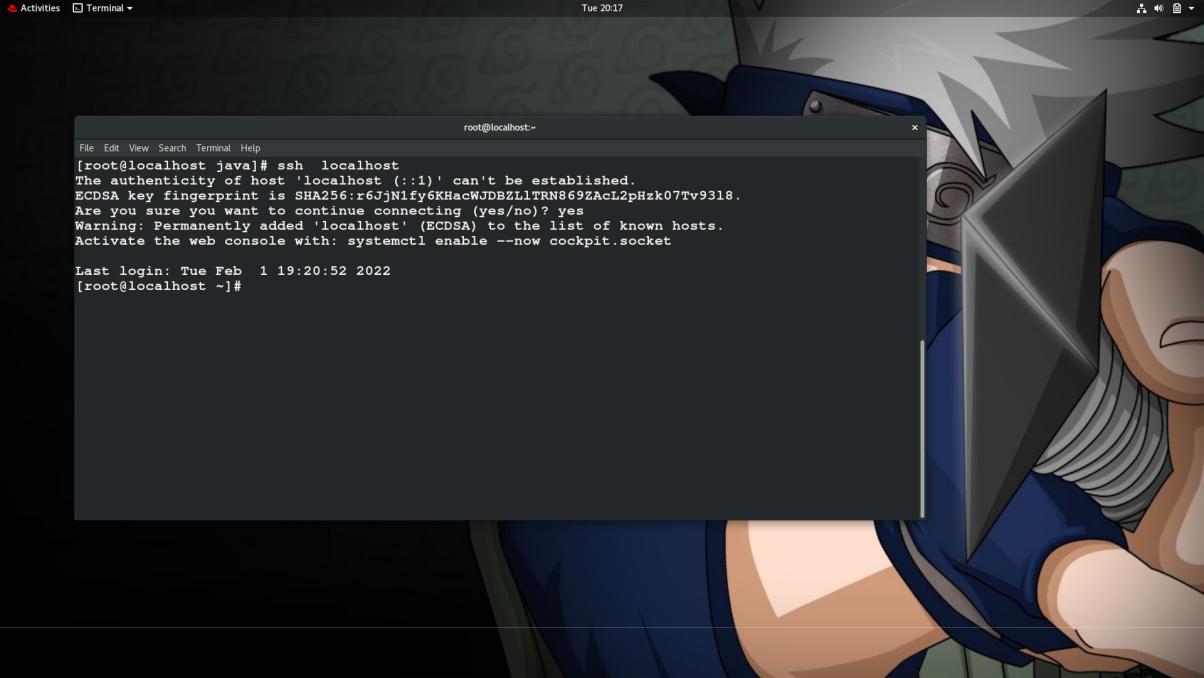
**Enable Passwordless SSH for Hadoop User**

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

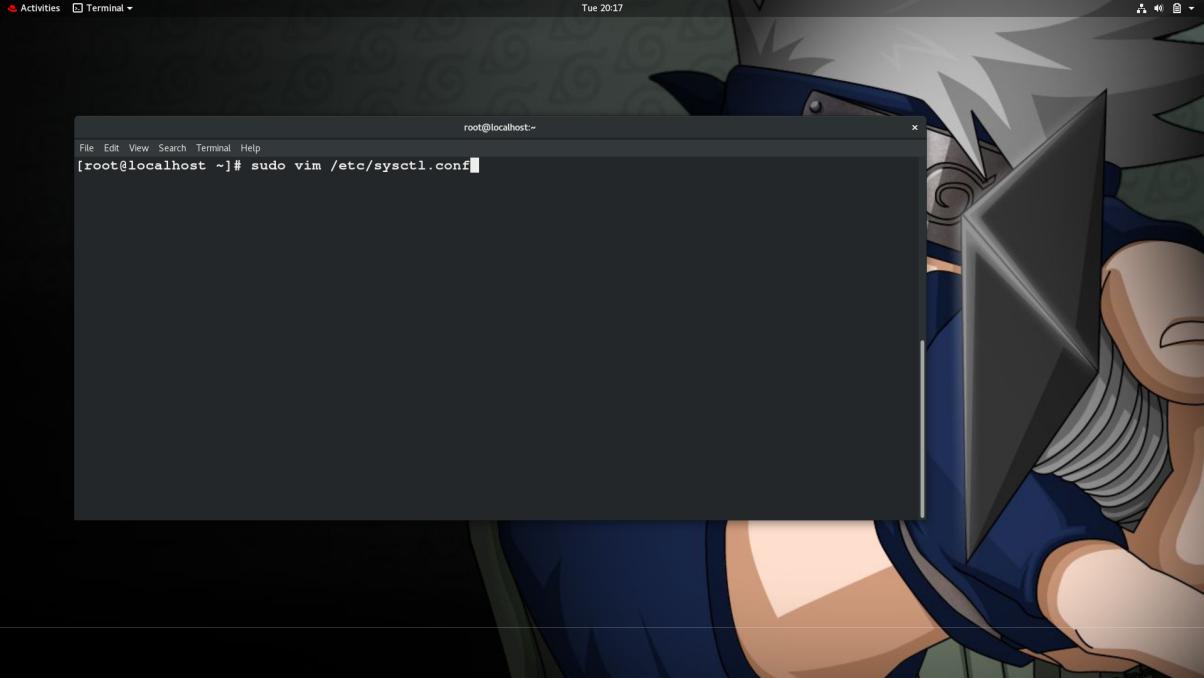
**cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys**

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

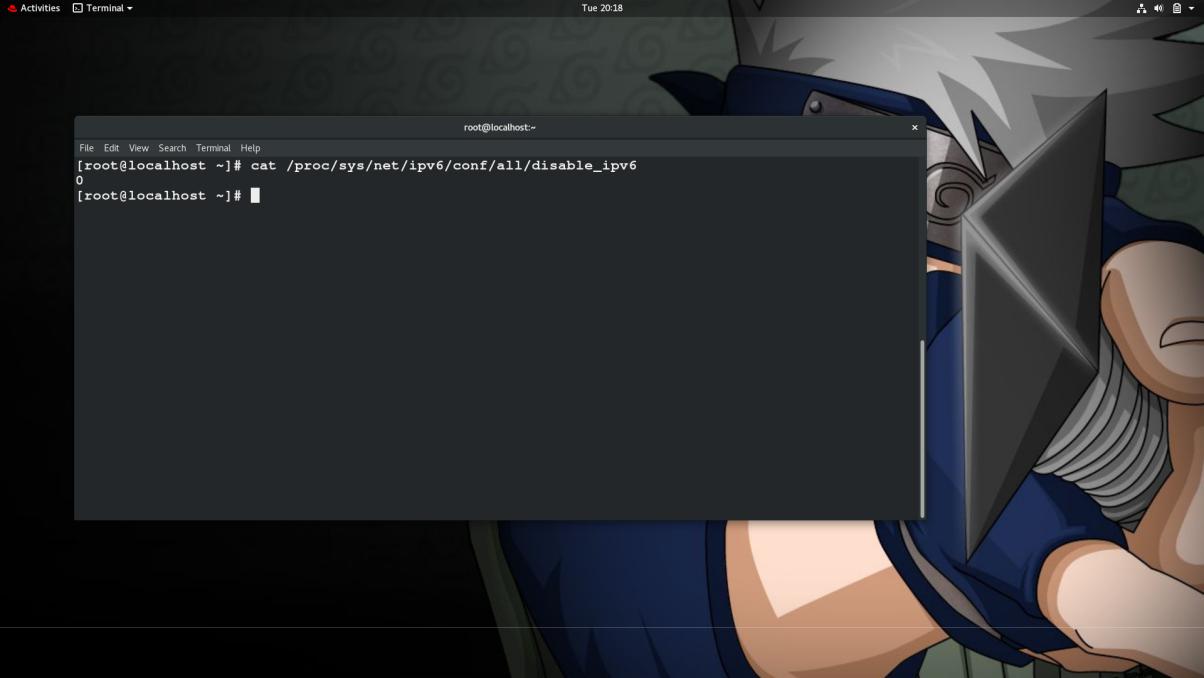
Use the cat command to store the public key as authorized\_keys in the *ssh* directory:

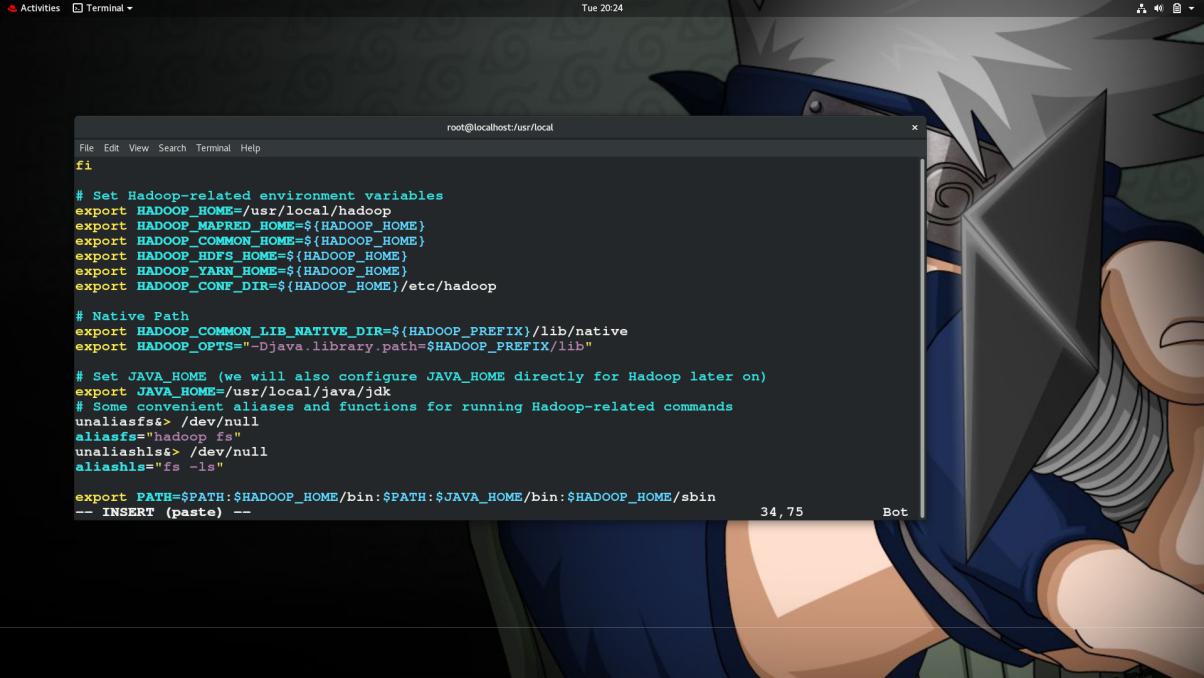
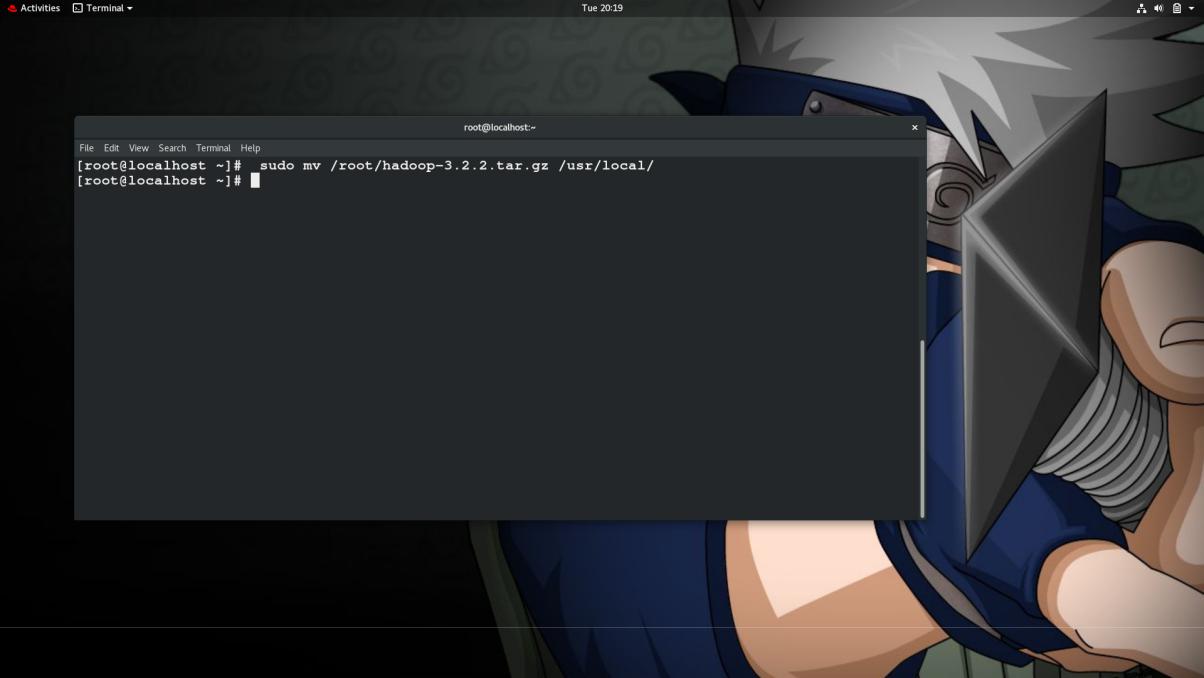


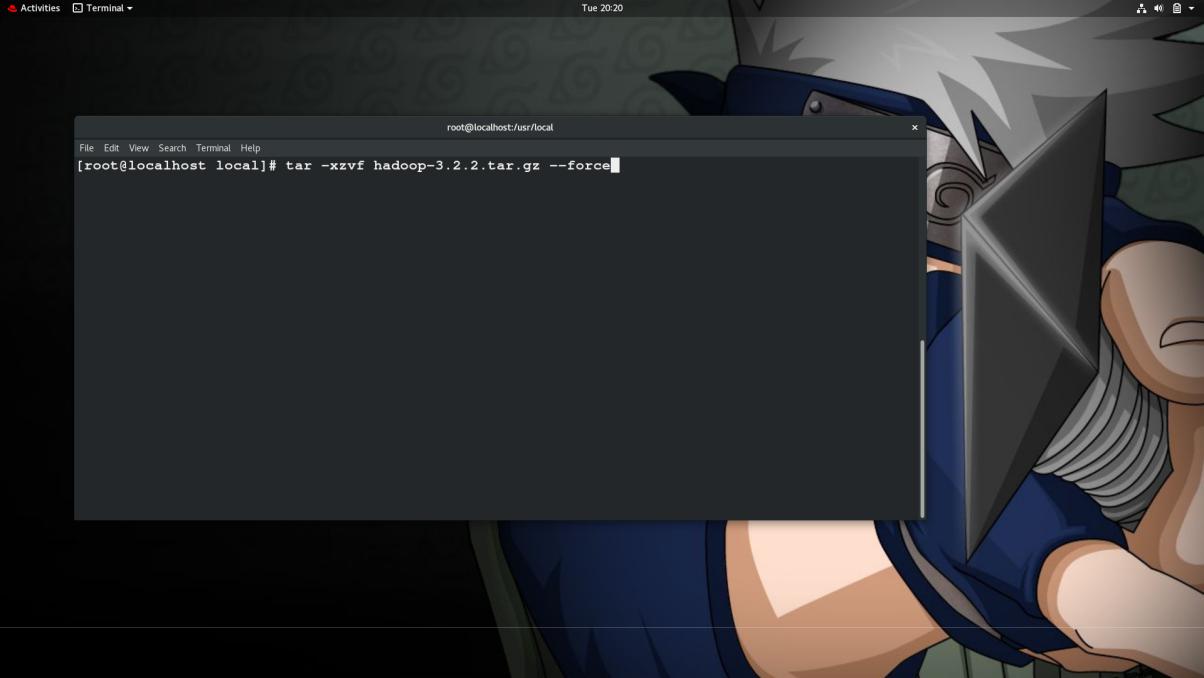
We will also disable the ipv6 and only use the ipv4 in the machine



**to crosscheck we can use cat over /proc/sys/net/ipv6/conf/all/disable\_ipv6 file**



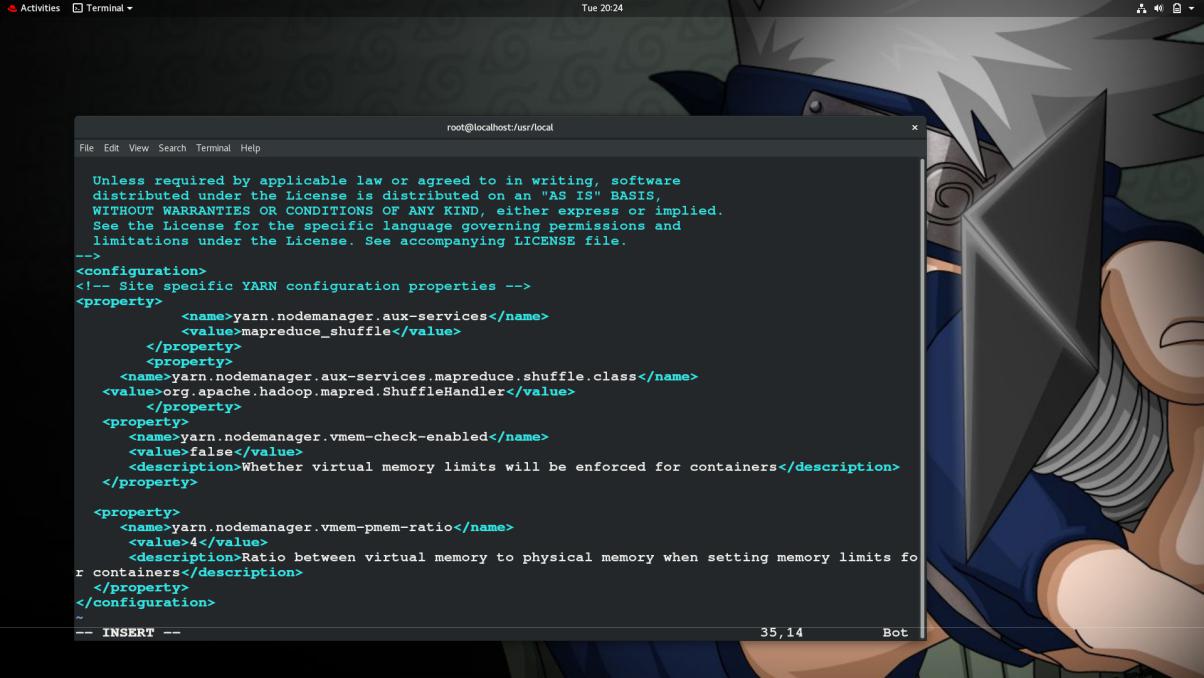
Move hadoop tar file from ~ file to /usr/local



Now its time for the hadoop files

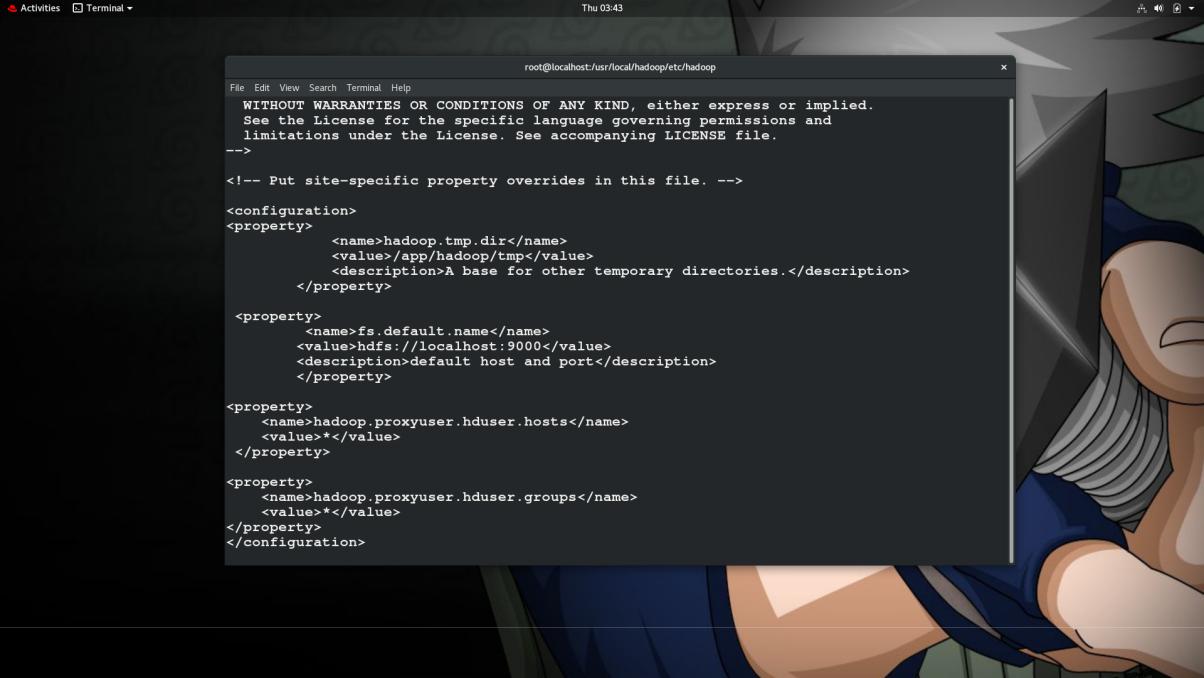
Configuration Changes in yarn-site.xml file

Edit **yarn-site.xml** with the following entries.



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.



\

**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.



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.

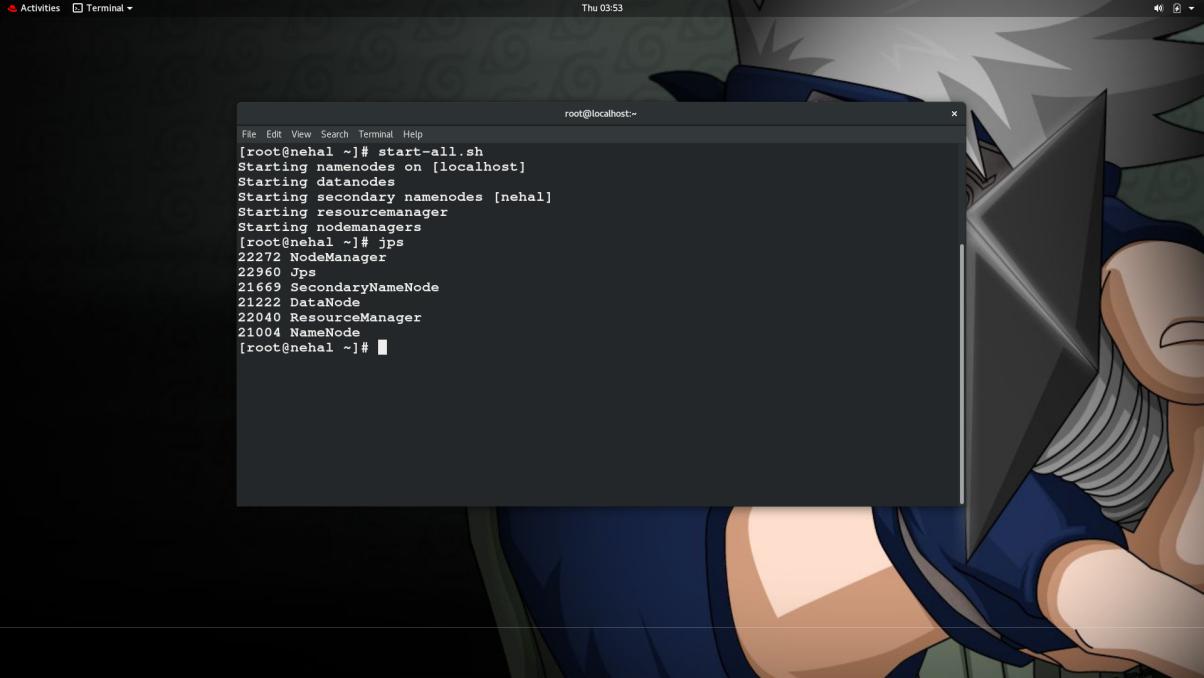


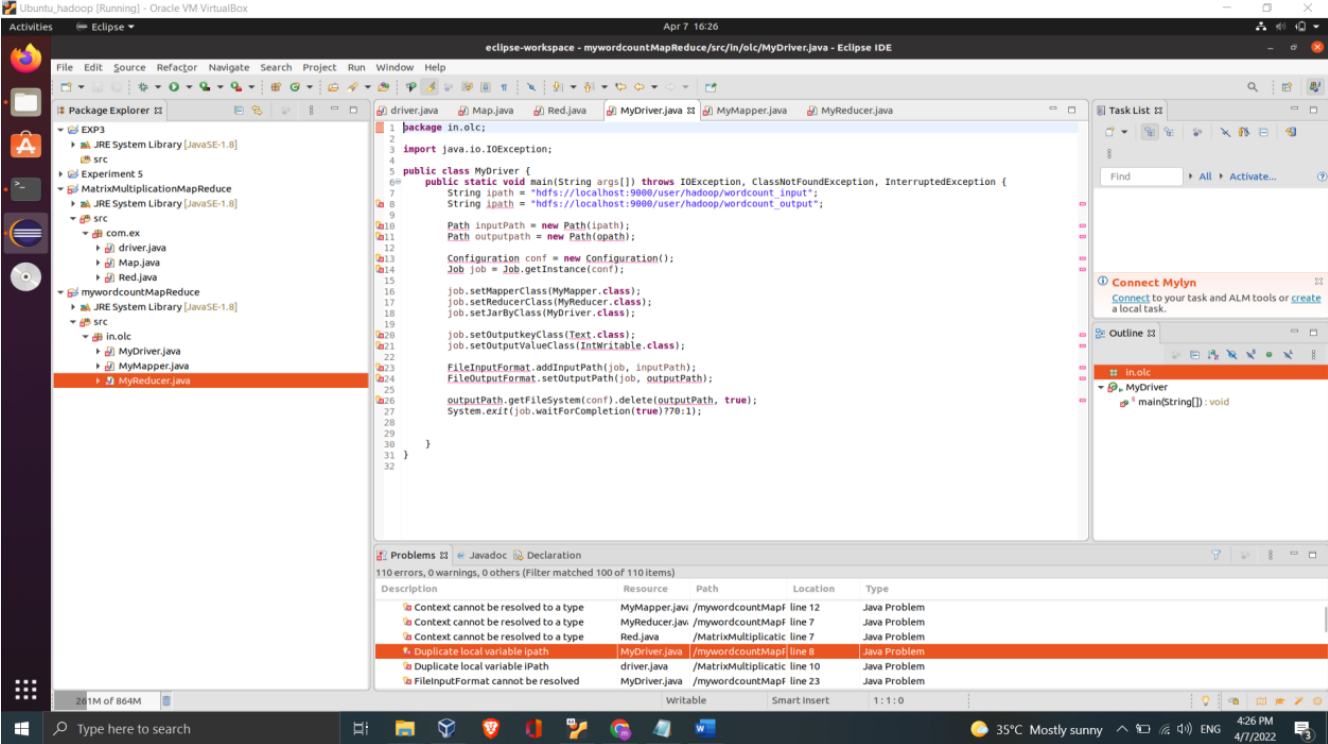
Once the Namenode has been formatted then start the HDFS using the

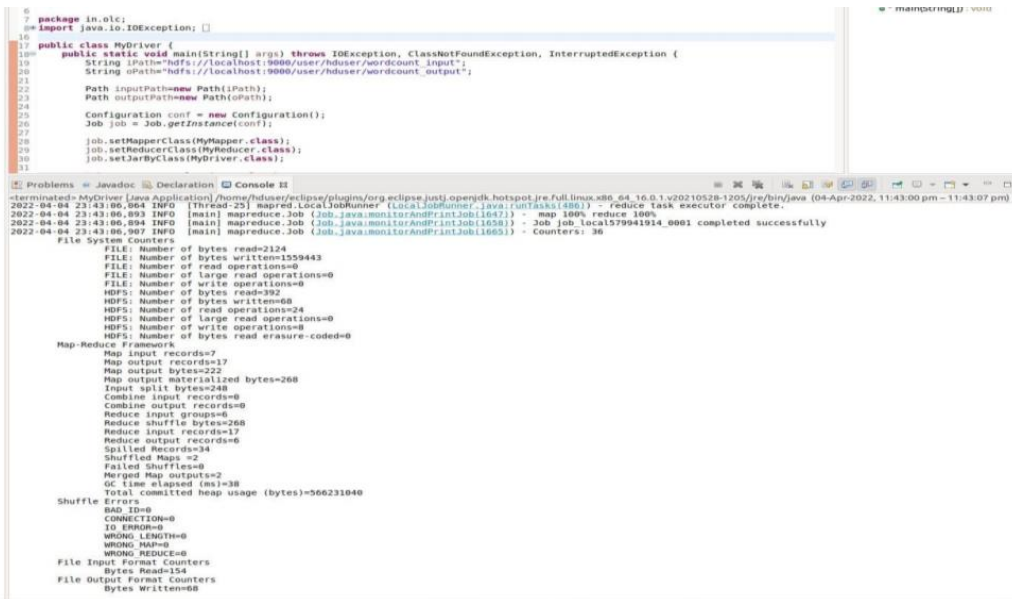
$ start-all.sh

All Services started successfully and all the node are

Working







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