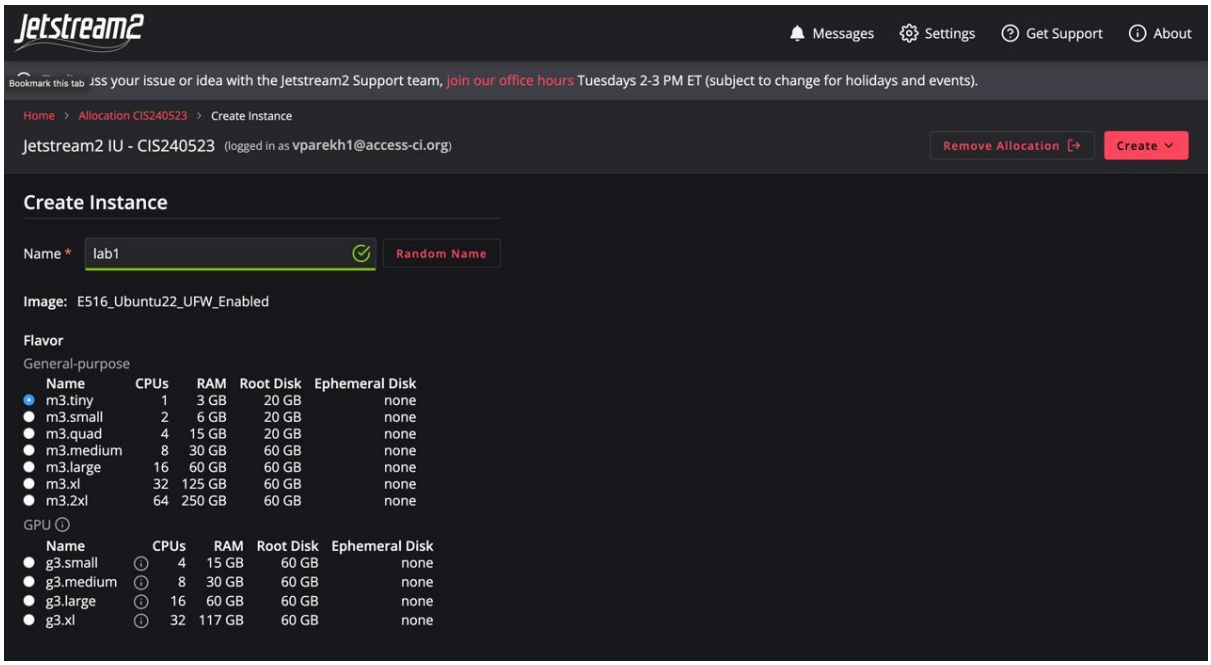
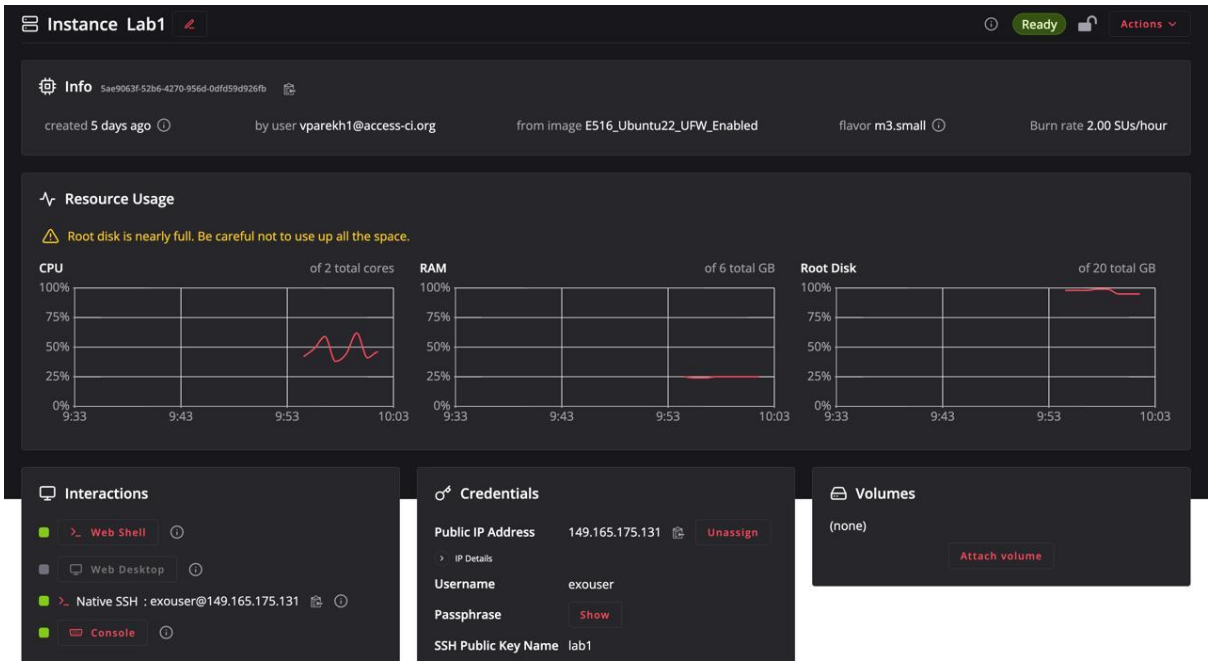
ECC Lab-01

Name: Dev Jatinbhai Patel

Vraj Parekh

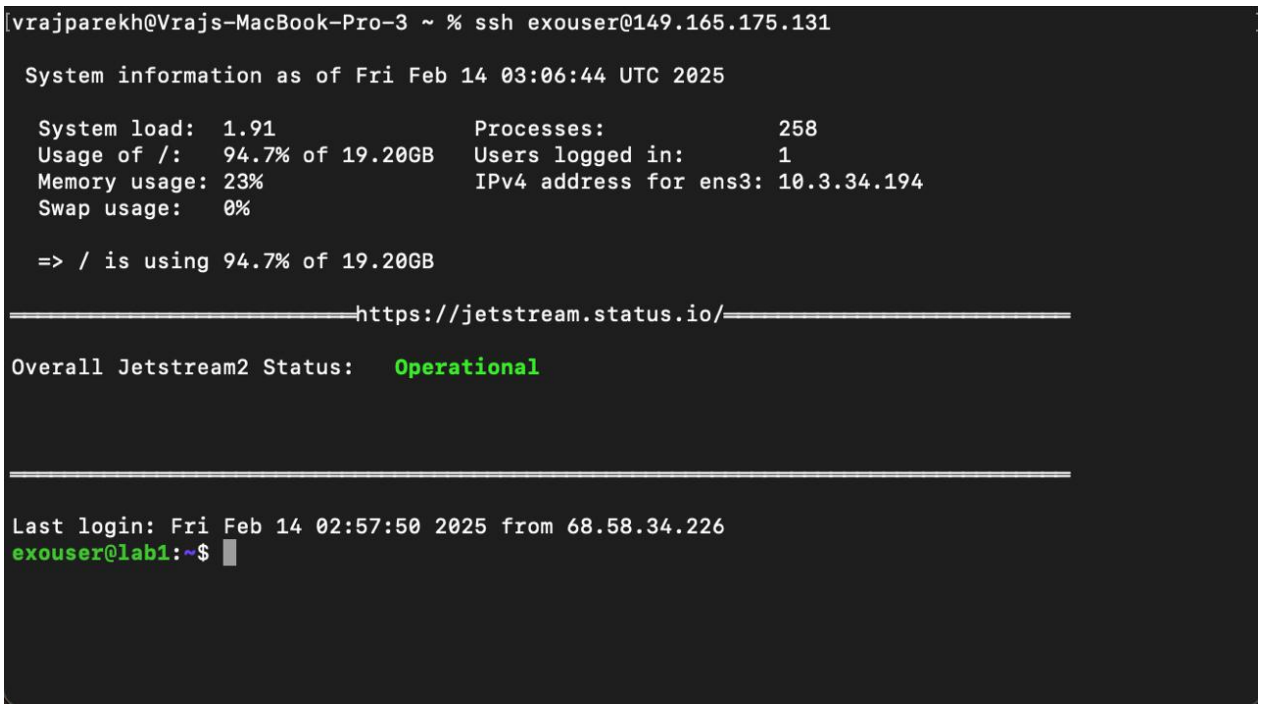
1.Create a cluster of 1 `tiny` instance from the virtual image called “E516 Ubuntu22 UFW Enabled”.



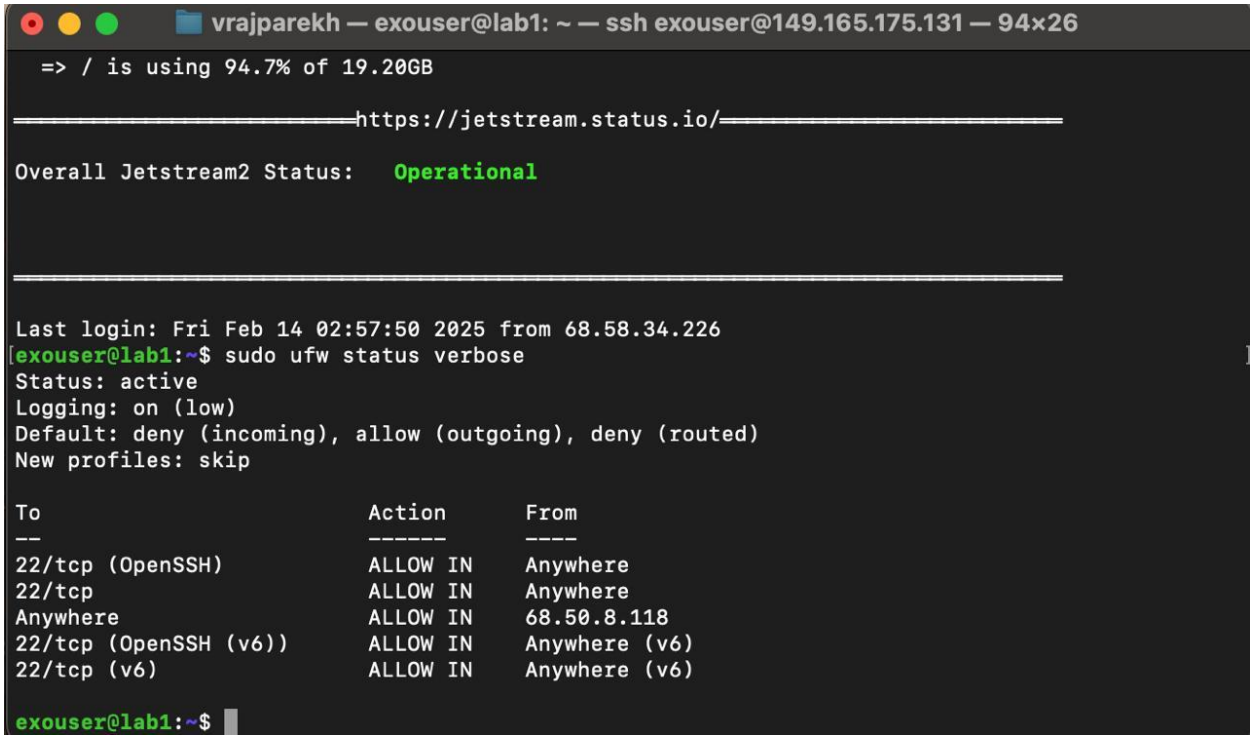


2.Configure the instance by the following steps:
a) Check if your UFW is working or not (super important!).
E.g., `sudo ufw status verbose`.
(see `https://linuxize.com/post/how-to-setup-a-firewall-with-ufw-on-ubuntu-20-04/`)
Warning: Disabling firewall or allowing unknown IP addresses to access your cluster will
lead to a 5-10 points deduction (Jetstream2 has a security monitor program checking all
instances' firewall constantly). In addition, the Jetstream2 Administrators will delete all
your instances.
b) Configure SSH logins using SSH key-pair so that your instanncs will not ask for pass-
words

Answer 2a.

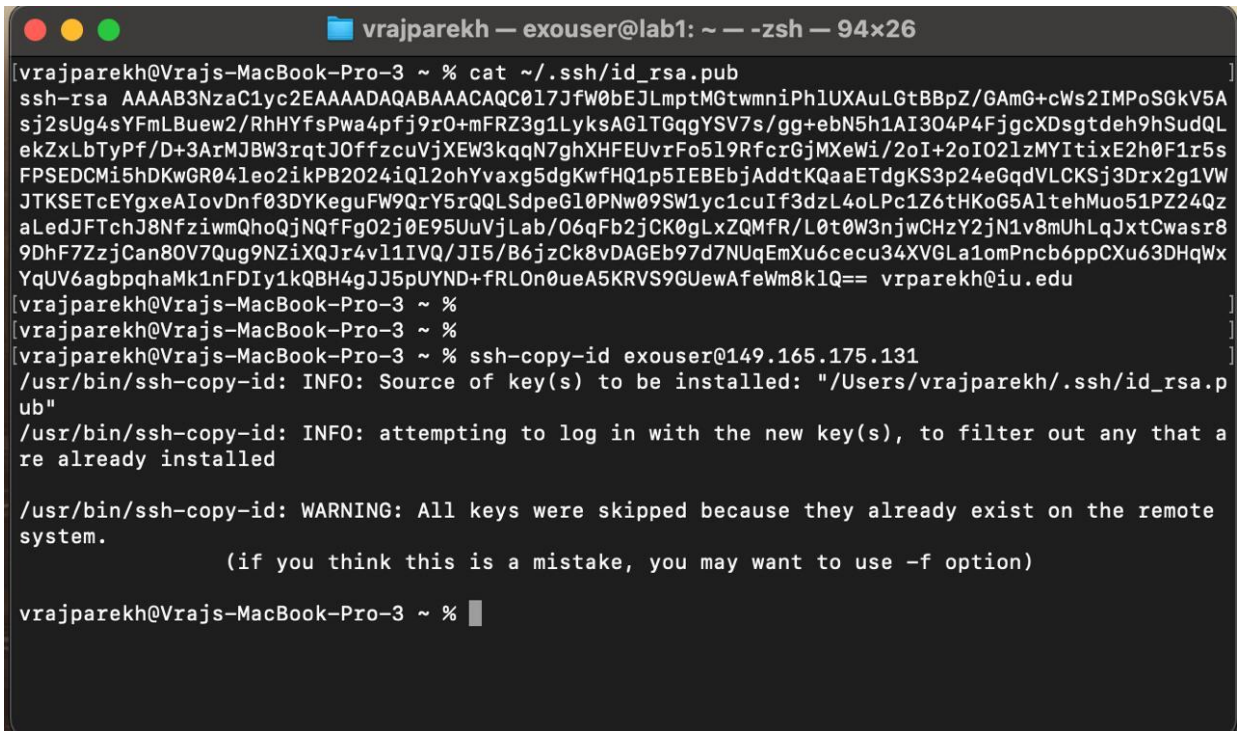


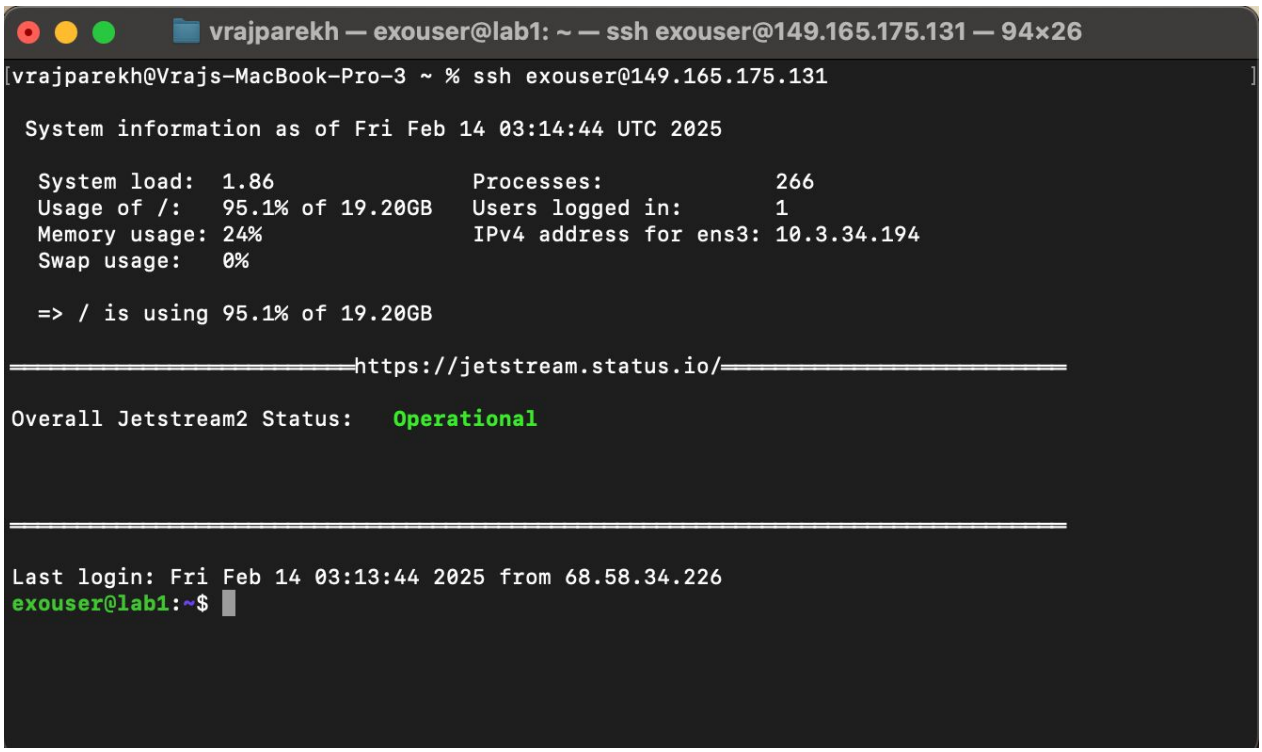
After this, we run the command to authorise the PC so that it can get through firewall.



2b.

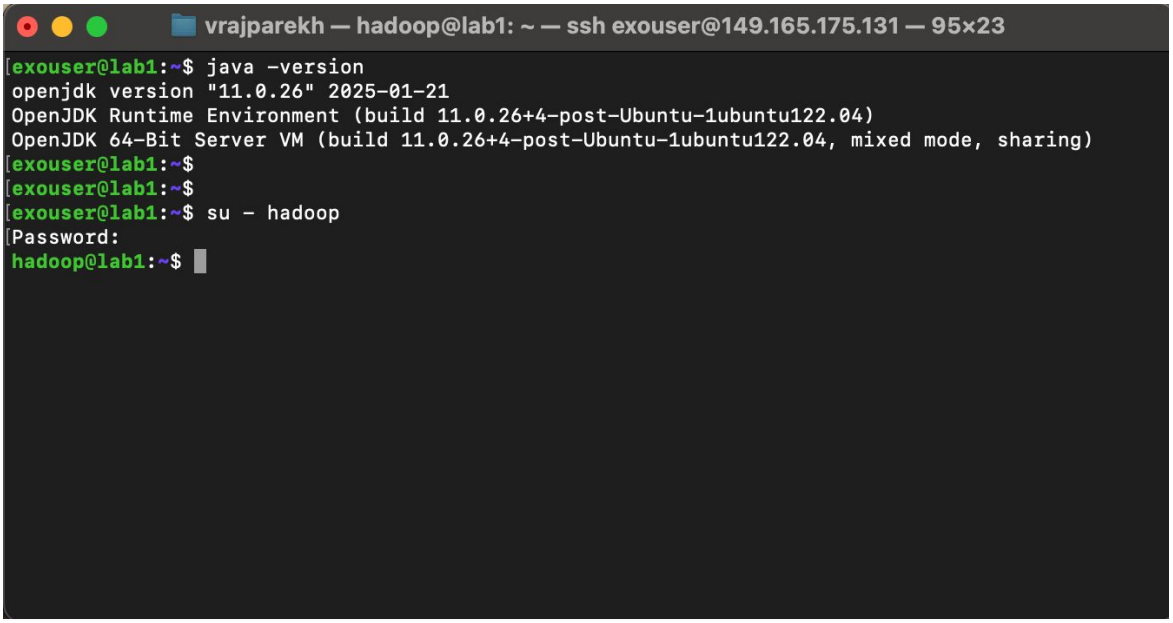
Then, we generated the ssh keys and below is generated key which will enable us to log-in wothout credentials, the snapshot of which is attached below:



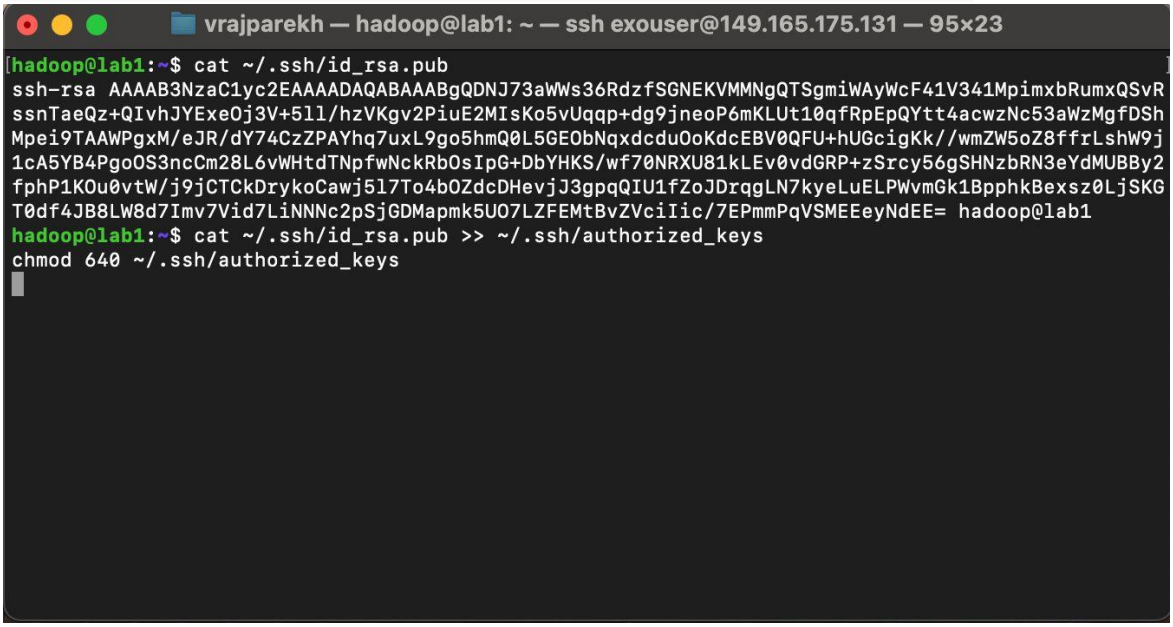


3. **Install Apache Hadoop, and test if your HDFS is working well or not. Test commands like `hdfs dfs -mkdir`. Please refer to `https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html` to see more `hdfs dfs` commands**

We were having the Java already installed, so that we directly created hadooop user through this command `sudo adduser hadoop`

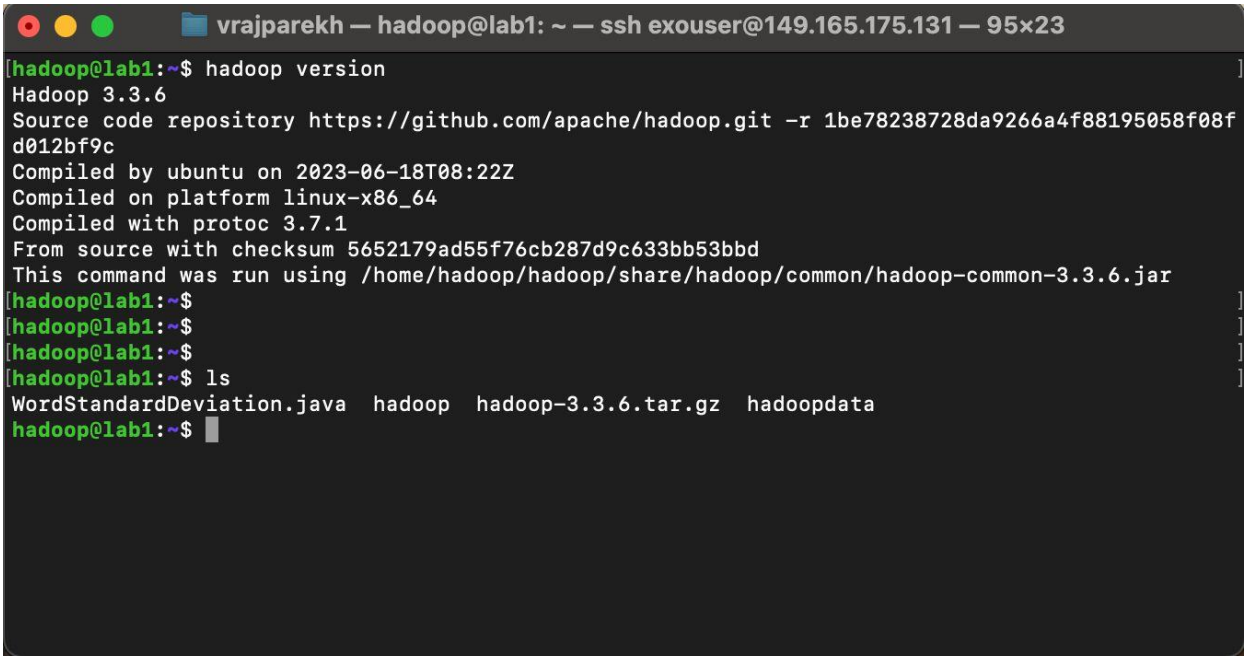


Then we configured the ssh keys to bypass authentication.



Then, we installed the hadoop in that user (named hadoop) using this command: `wget` https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz

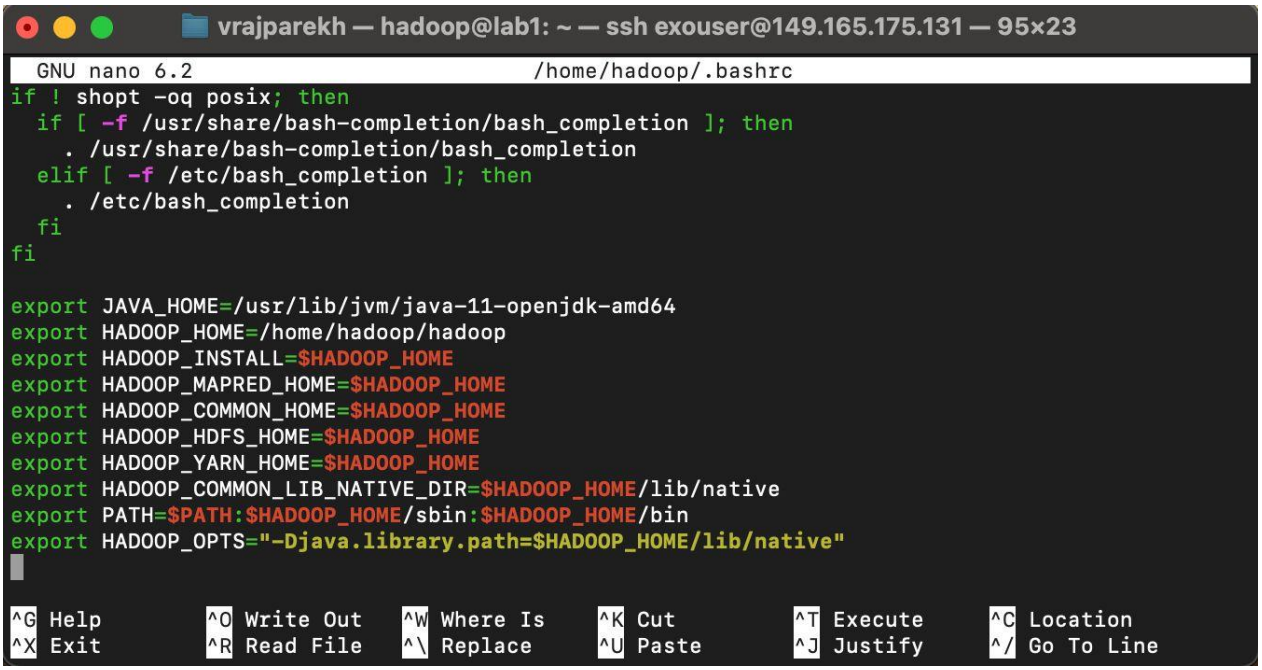
```
tar -xvzf hadoop-3.3.6.tar.gz
```

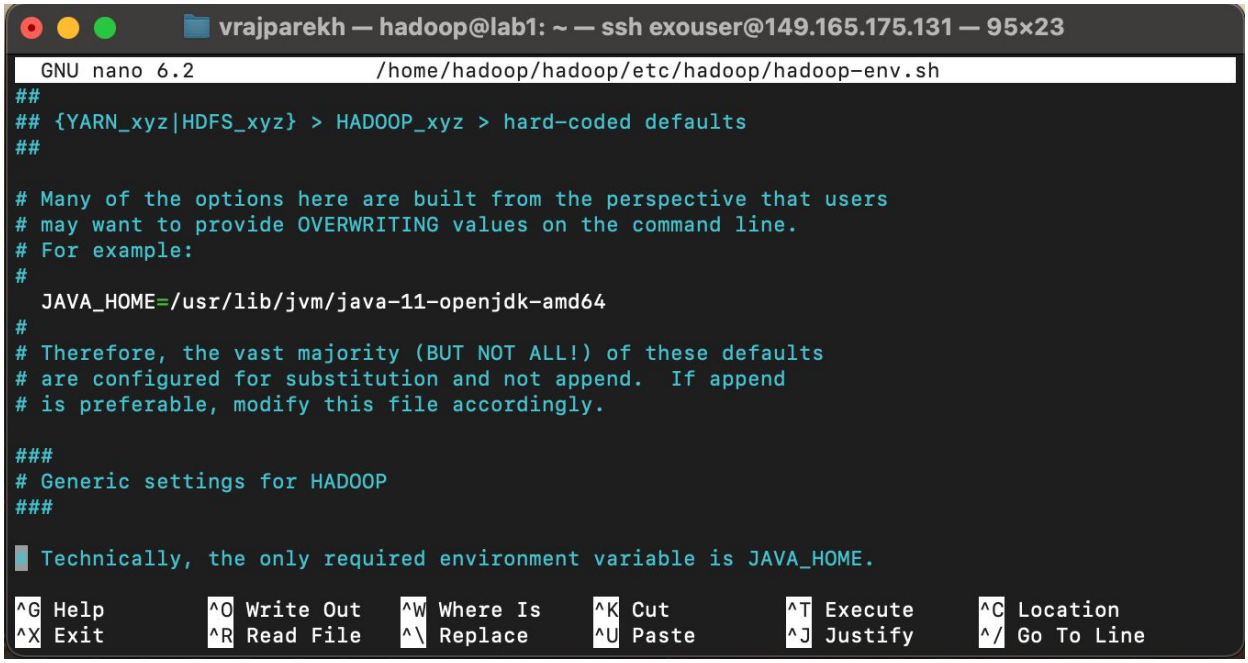


4. Configure the MapReduce programming framework properly.

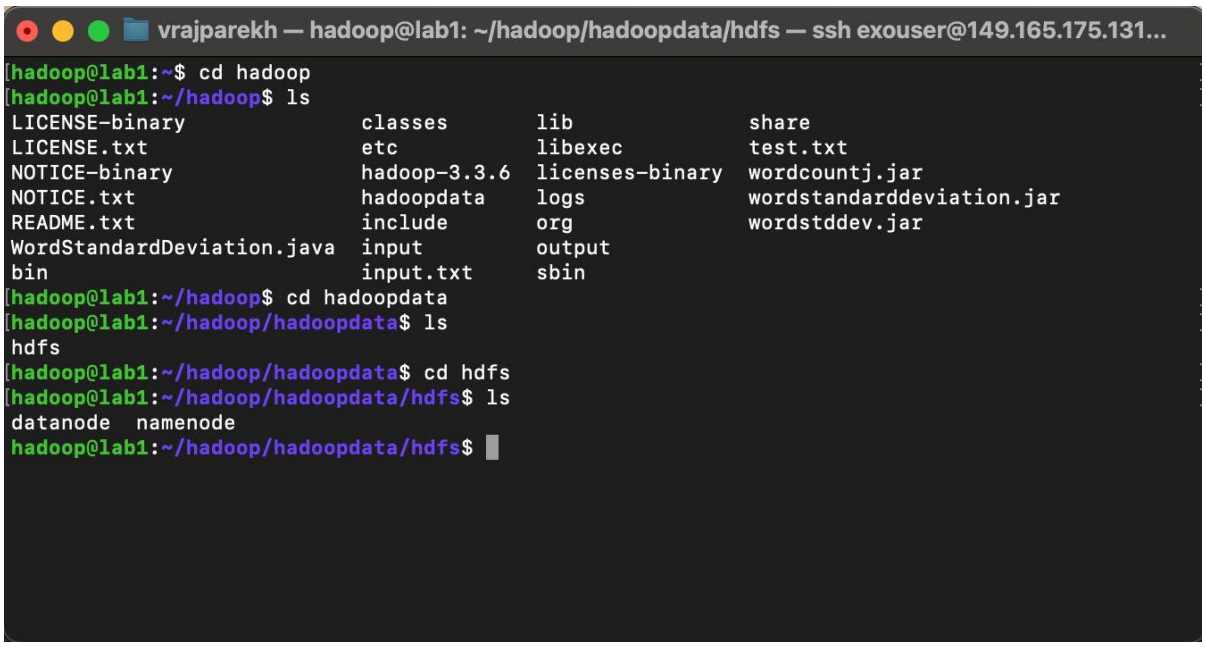
Answer:

After inftalling the hadoop, we made changes in the bashrc and env.sh files which are below:

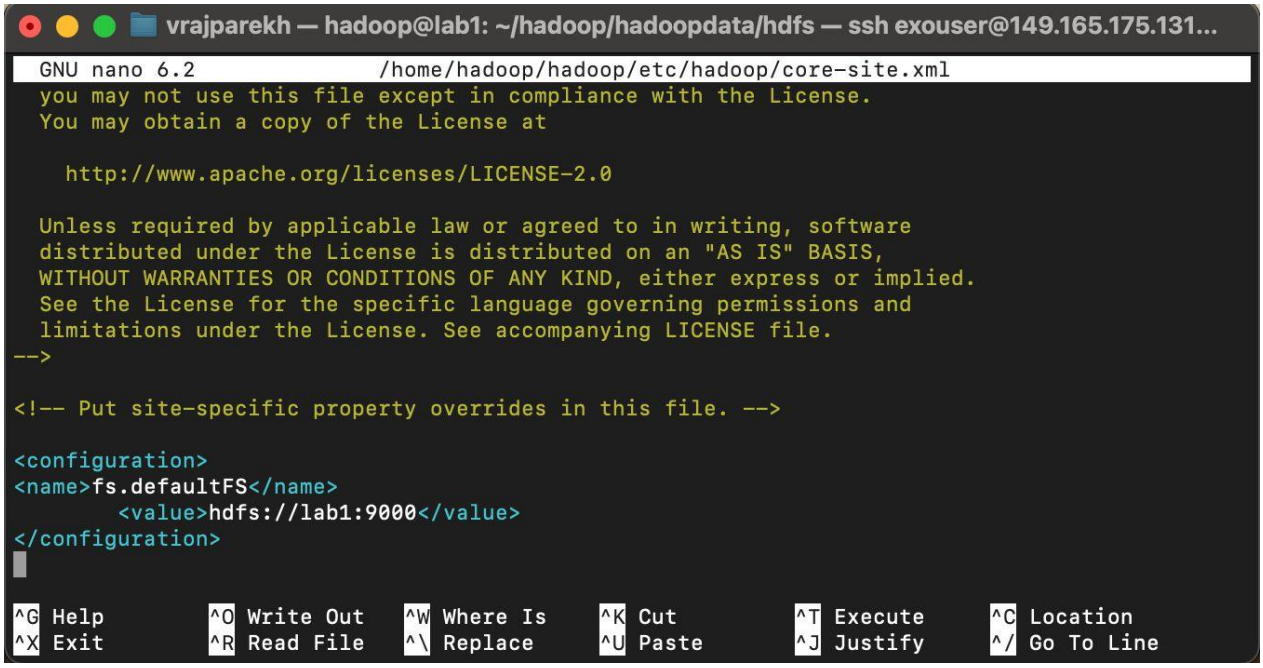




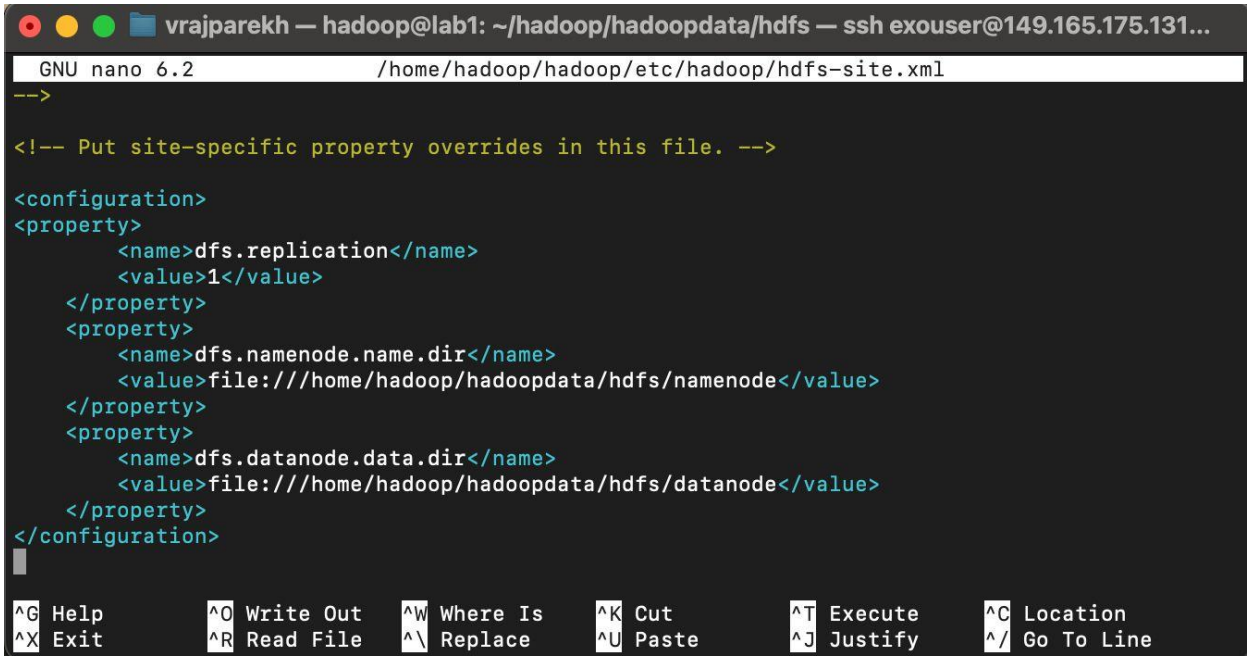
After this, we configures hadoop by creating the namenode and datanode directory.



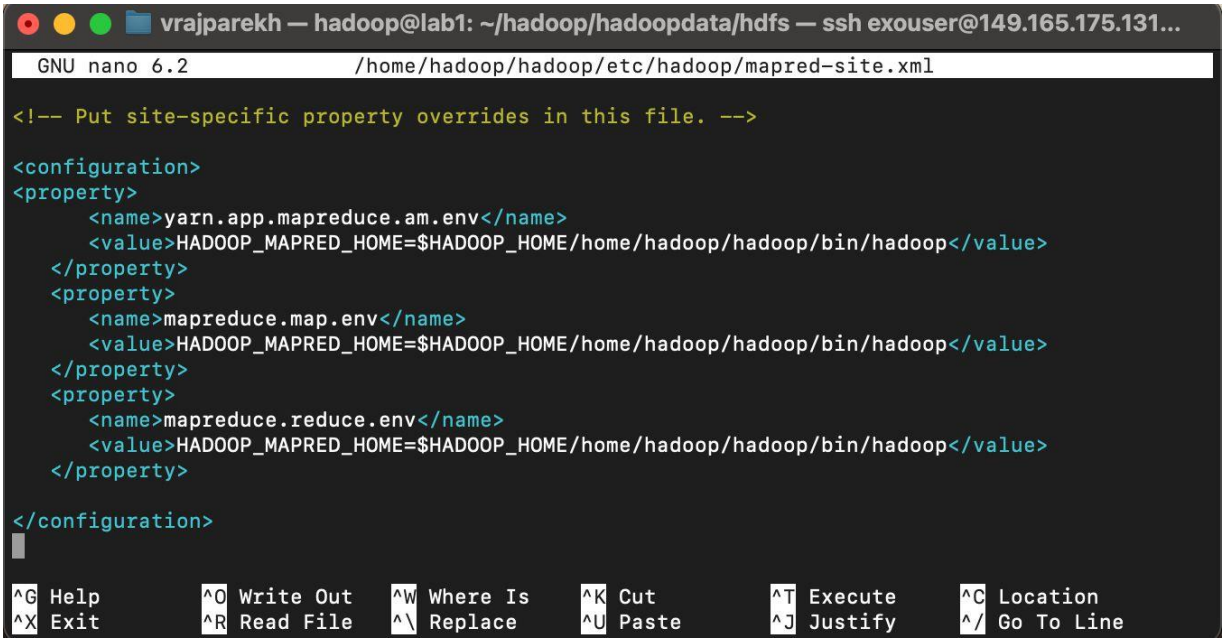
We also made changes in the core-site.xml which is shown below:

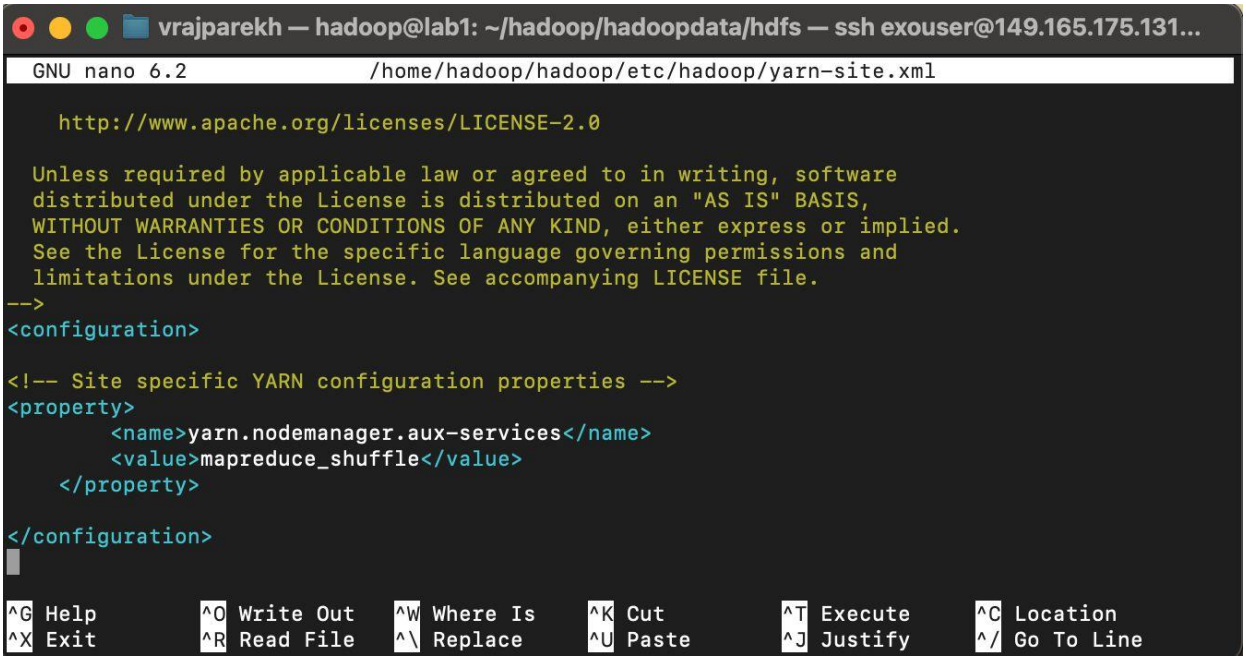


After completion of the changes in core-site, we made changes in hdfs-site.xml:

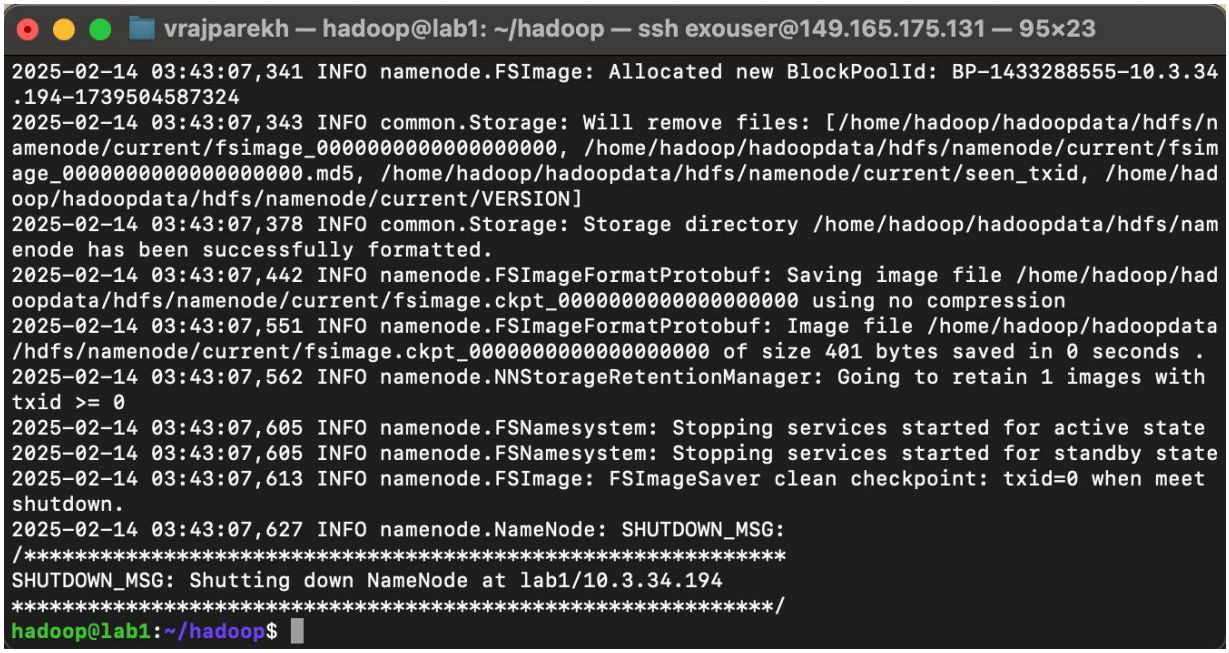


Following that, we also changed yarn.xml and mapred.xml:

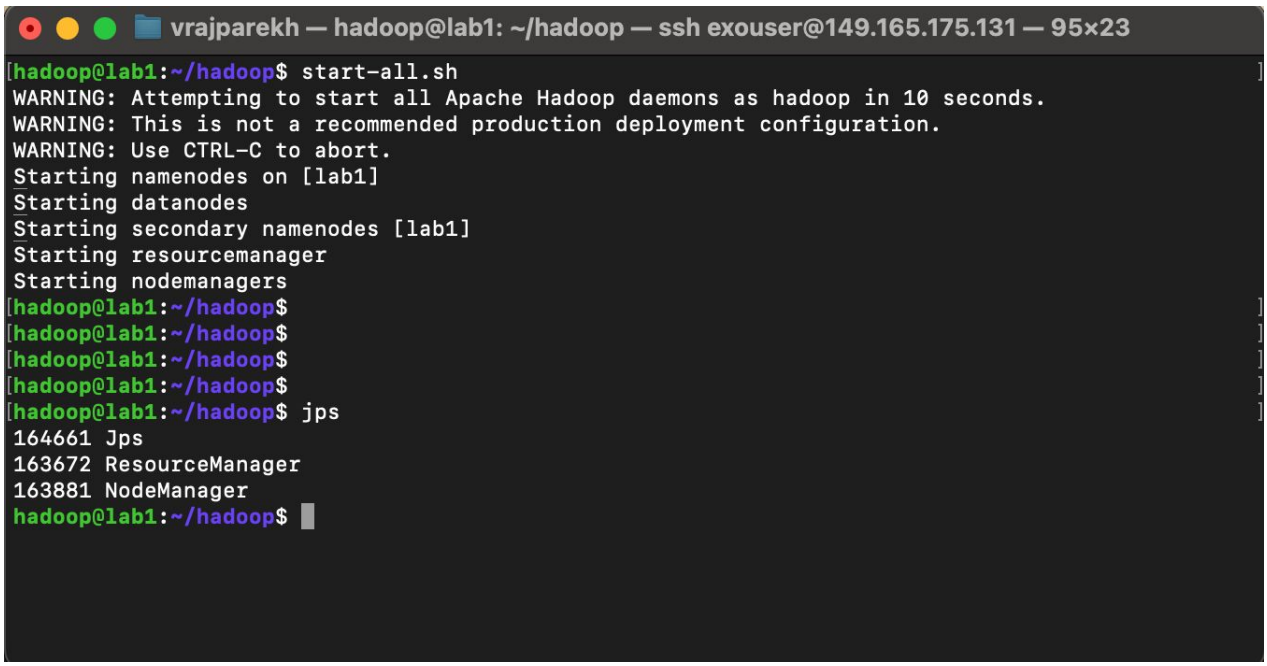




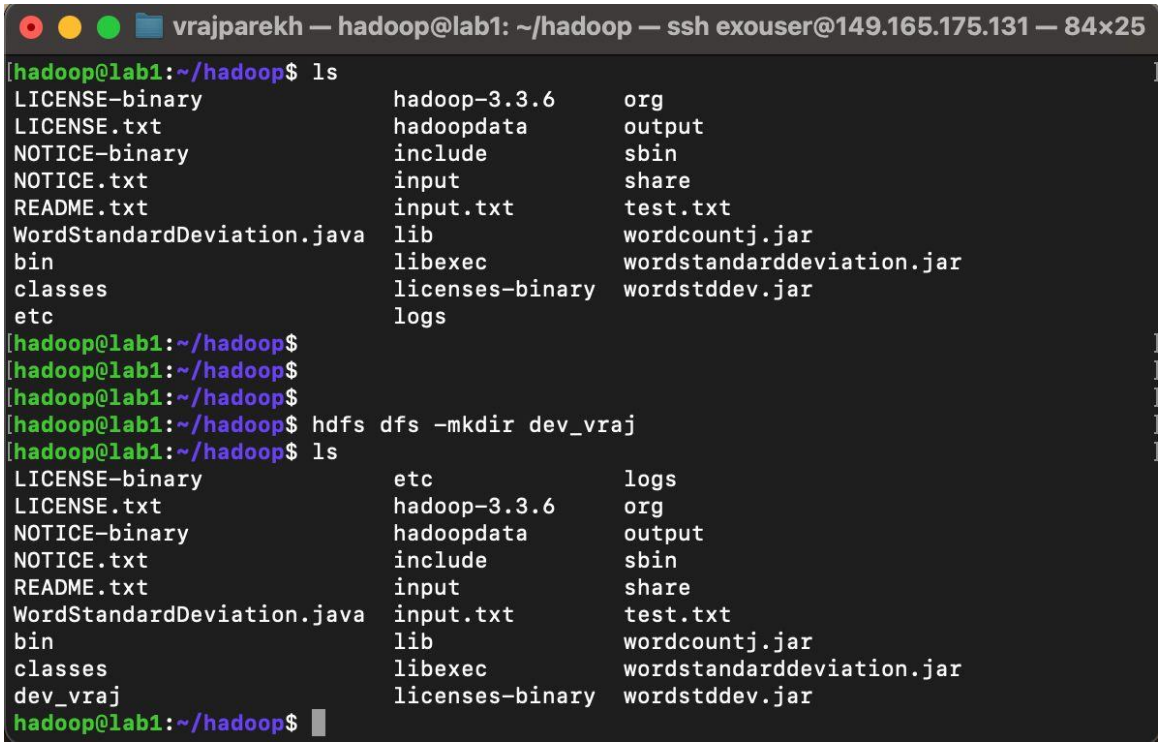
Finally, we are starting the hadoop cluster:



After that, we start the hadoop cluster and also checked all status of hadoop services:



Atlast, we tested commands like mkdir to check the working status:

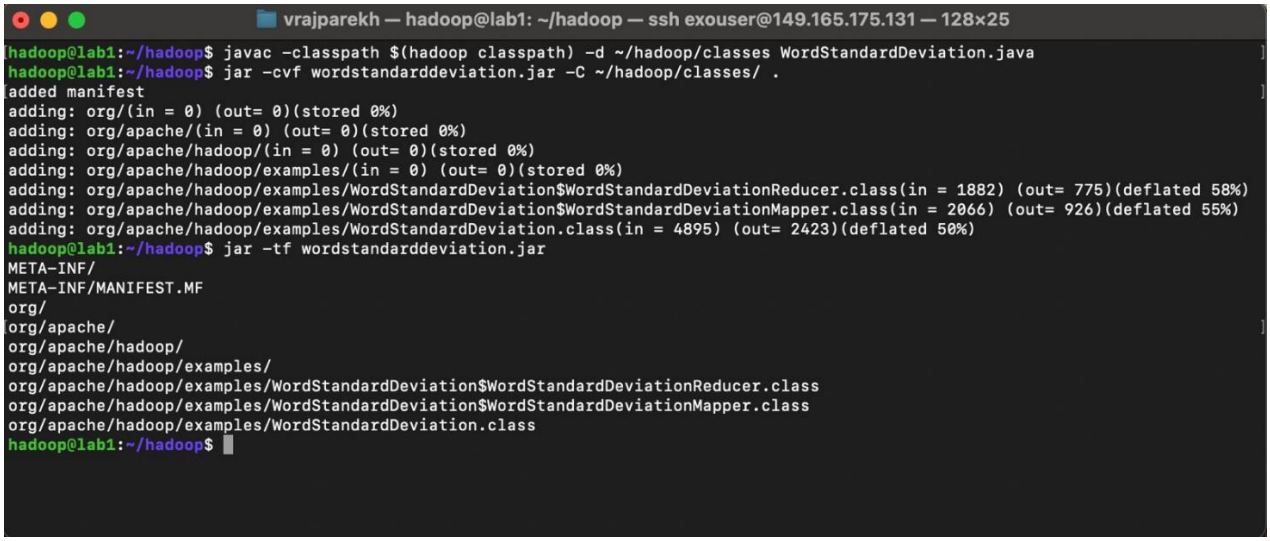


5 **Compile and run the MapReduce example of `WordStandardDeviation.java` code.**
To save your time to find the file, the example's source code is posted on Canvas now.
Also, to help you learn how to compile and create a jar file, please refer to `https://engineering.tomtom.com/create-java-jar/`.

We uploaded the file through sftp and below is the snapshot of it:

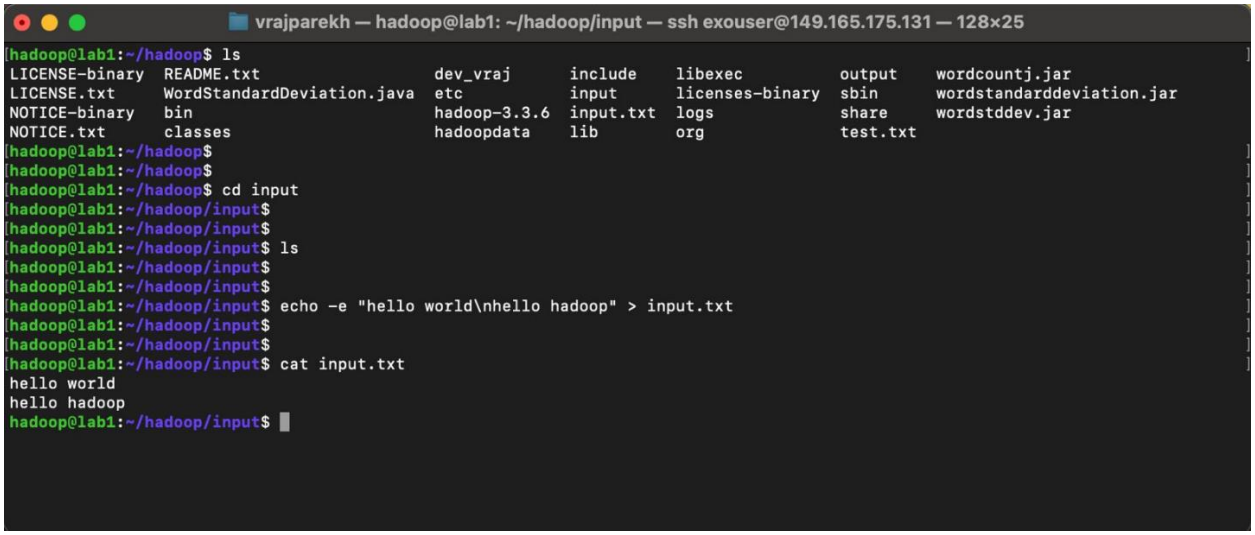
After that, we compiled the given .java file and then converted the created .class file into the .jar file using this command:

jar -cvf wordstandarddeviation.jar -C ~/hadoop/classes/

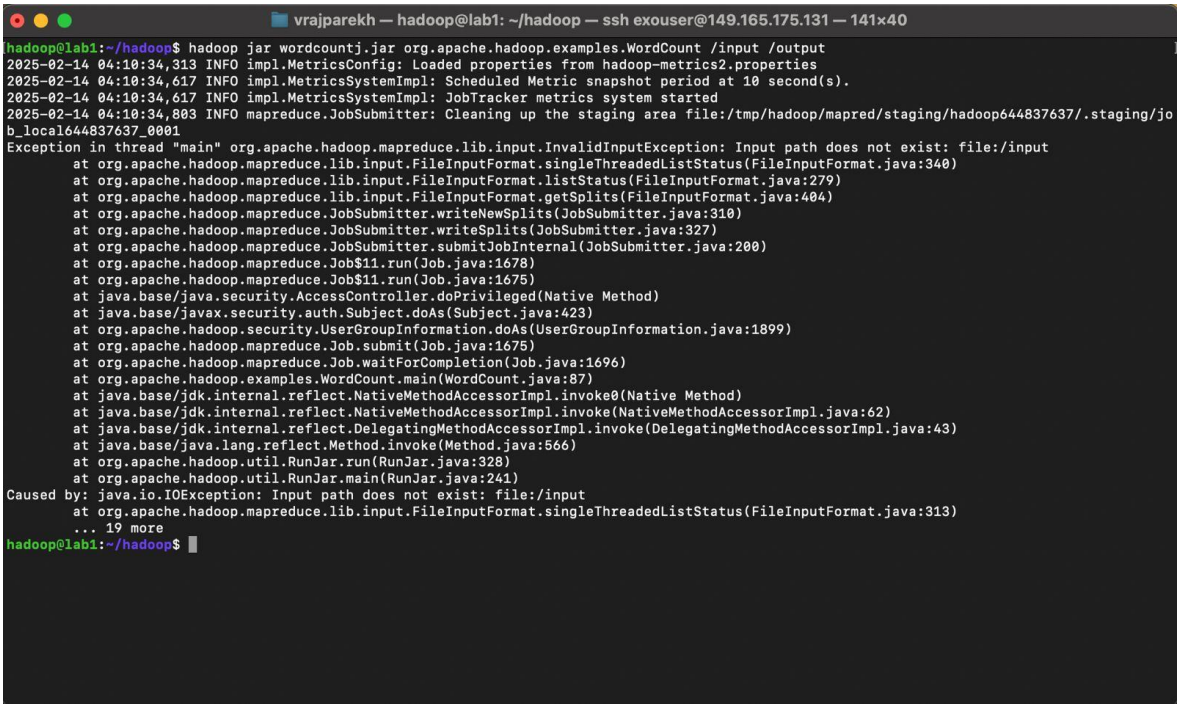


Then after, we created the input and output named directory using the command:

hdfs dfs -mkdir /input
hdfs dfs -mkdir /output



Then after, we run hadoop with mapreduce:



Here,is the final output result: