**Big data lab:**

**Ex 2 commands:**

[**Apache Hadoop 3.3.6 Installation on Ubuntu 22.04 | by Abhik Dey | Medium**](https://medium.com/@abhikdey06/apache-hadoop-3-3-6-installation-on-ubuntu-22-04-14516bceec85)

sudo apt update && sudo apt install openjdk-8-jdk

java -version

sudo apt install ssh

sudo adduser Hadoop

su – Hadoop

ssh-keygen -t rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys   
chmod 640 ~/.ssh/authorized\_keys

ssh localhost

su – hadoop

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

mv hadoop-3.3.6 hadoop

nano ~/.bashrc

* Append the below lines to the file.

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
export HADOOP\_HOME=/home/hadoop/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 HADOOP\_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"

source ~/.bashrc

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

append - JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

cd hadoop/

mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}

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

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

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

<configuration>  
 <property>  
 <name>dfs.replication</name>  
 <value>1</value>  
 </property>  
 <property>  
 <name>dfs.namenode.name.dir</name>  
 <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>  
 </property>  
 <property>  
 <name>dfs.datanode.data.dir</name>  
 <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>  
 </property>  
 </configuration>

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

<configuration>  
 <property>  
 <name>yarn.app.mapreduce.am.env</name>  
 <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME/home/hadoop/hadoop/bin/hadoop</value>  
 </property>  
 <property>  
 <name>mapreduce.map.env</name>  
 <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME/home/hadoop/hadoop/bin/hadoop</value>  
 </property>  
 <property>  
 <name>mapreduce.reduce.env</name>  
 <value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME/home/hadoop/hadoop/bin/hadoop</value>  
 </property>  
</configuration>

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

<configuration>  
 <property>  
 <name>yarn.nodemanager.aux-services</name>  
 <value>mapreduce\_shuffle</value>  
 </property>  
</configuration>

hdfs namenode -format

start-all.sh

jps

sudo apt install net-tools

ifconfig

Namenode - <http://hadoop.tecadmin.net:9870/>

Resource manager- <http://hadoop.tecadmin.net:8088/>

Verify Hadoop cluster-

hdfs dfs -mkdir /test1  
hdfs dfs -mkdir /logs

hdfs dfs -ls /

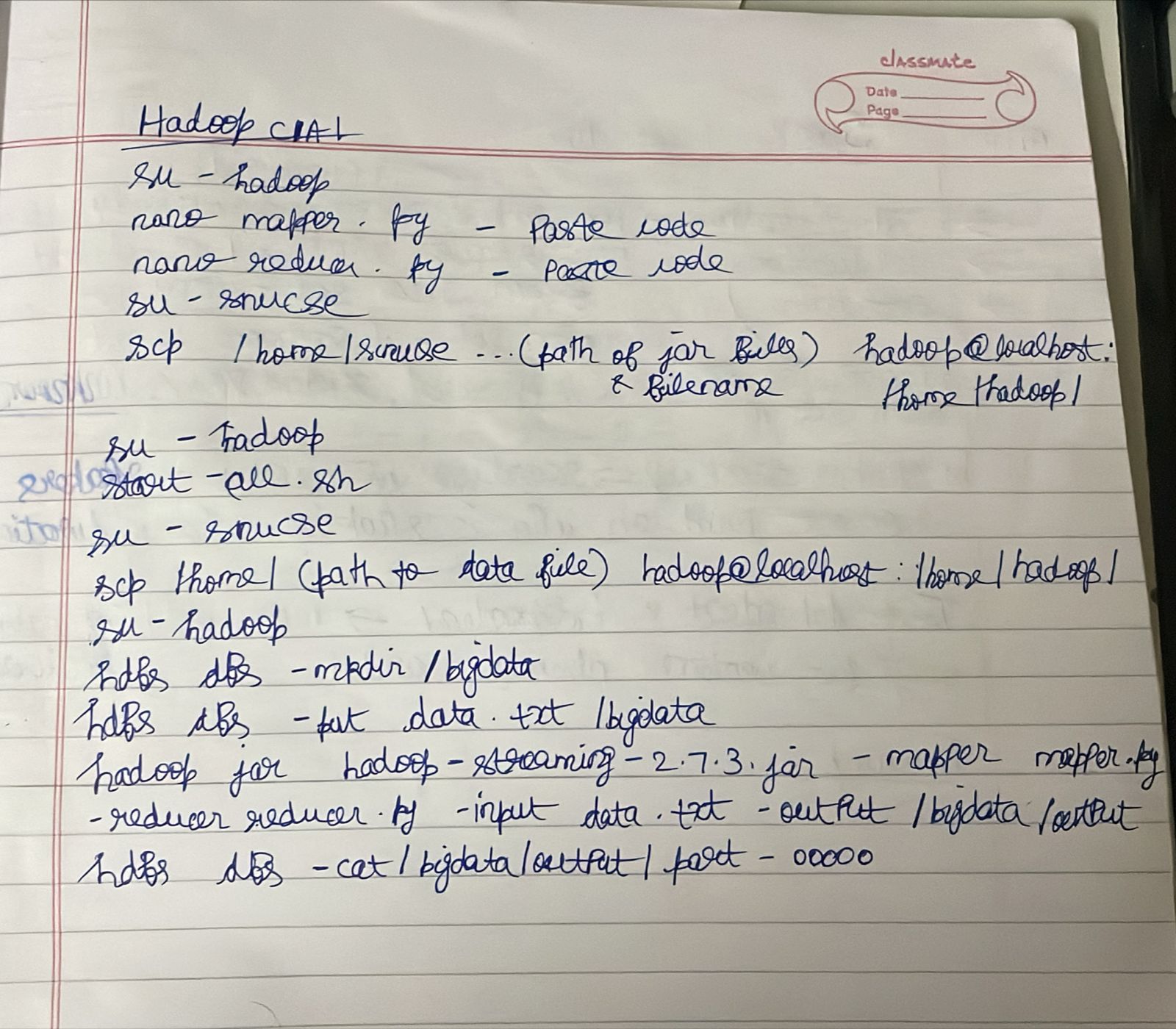
hdfs dfs -put /var/log/\* /logs/

stop-all.sh

**EX-3:**

Implement a simple map-reduce code for the wordcount problem using Java/Python. (Create the jar files and run the code using HDFS.)

<https://www.geeksforgeeks.org/hadoop-streaming-using-python-word-count-problem/>





hdfs dfs -mkdir /lab\_cia\_1

hdfs dfs -put /mnt/c/Users/navin/test\_python\_hadoop/lab\_cia\_1/CRND0103-2024-VA\_Charlottesville\_2\_SSE.txt /lab\_cia\_1

hdfs dfs -ls /lab\_cia\_1

chmod 777 mapper.py reducer.py

hadoop jar /mnt/c/Users/navin/test\_python\_hadoop/lab\_cia\_1/hadoop-streaming-2.7.3.jar -input /lab\_cia\_1/CRND0103-2024-VA\_Charlottesville\_2\_SSE.txt -output /lab\_cia\_1/output -mapper /mnt/c/Users/navin/test\_python\_hadoop/lab\_cia\_1/mapper.py -reducer /mnt/c/Users/navin/test\_python\_hadoop/lab\_cia\_1/reducer.py