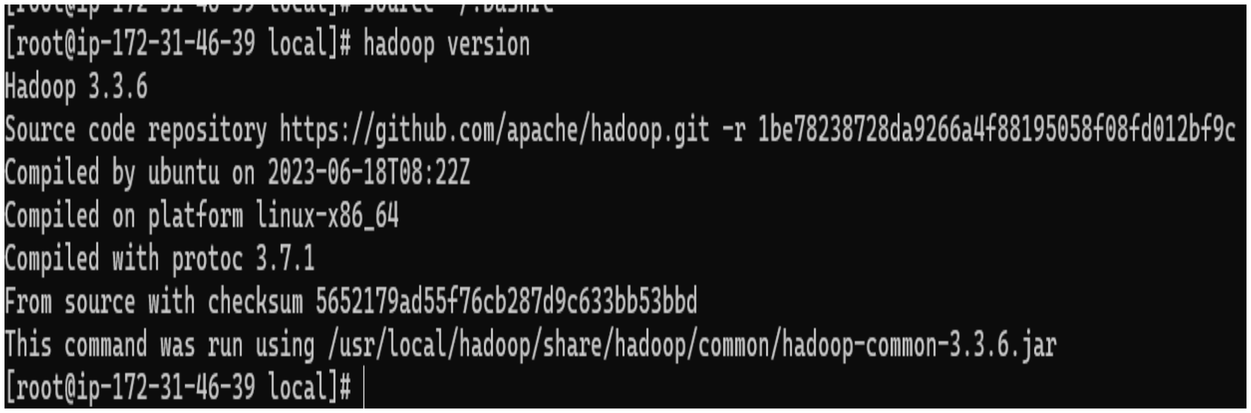
PROCEDURE: -

- Switch to superuser mode using sudo su.
- Update all packages with sudo yum update -y.
- Install Java with sudo yum install java-11-openjdk-devel -y or sudo yum install java-1.8.0-amazon-corretto-devel -y.
- Verify the Java installation using java -version.
- Navigate to /usr/local/ using cd /usr/local/.
- Download Hadoop with sudo wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz.
- Extract the Hadoop archive using sudo tar -xvzf hadoop-3.3.6.tar.gz.
- Rename the extracted folder with sudo mv hadoop-3.3.6 hadoop.
- Edit the .bashrc file to add Hadoop and Java environment variables.
- Reload the .bashrc file with source ~/.bashrc and verify Hadoop installation using hadoop version.

CODING: -

- sudo su
- sudo yum update -y
- sudo yum install java-11-openjdk-devel -y **or** sudo yum install java-1.8.0-amazon-corretto-devel -y
- java -version
- cd /usr/local/
- sudo wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
- sudo tar -xvzf hadoop-3.3.6.tar.gz
- sudo mv hadoop-3.3.6 hadoop
- sudo nano ~/.bashrc
    - # Hadoop variables
    - export HADOOP_HOME=/usr/local/hadoop
    - export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
    - # Java variables
    - export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:/bin/java::")
    - export PATH=$PATH:$JAVA_HOME/bin
- source ~/.bashrc
- hadoop version

OUTPUT: -



```
[root@ip-172-31-46-39 local]# source ~/.bashrc
[root@ip-172-31-46-39 local]# hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
[root@ip-172-31-46-39 local]#
```

PROCEDURE: -

- Switch to superuser mode using sudo su.
- Update all packages with sudo yum update -y.
- Install Java with sudo yum install java-11-openjdk-devel -y or sudo yum install java-1.8.0-amazon-corretto-devel -y.
- Verify the Java installation using java -version.
- Navigate to /usr/local/ using cd /usr/local/.
- Download Hadoop with sudo wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz.
- Extract the Hadoop archive using sudo tar -xvzf hadoop-3.3.6.tar.gz.
- Rename the extracted folder with sudo mv hadoop-3.3.6 hadoop.
- Edit the .bashrc file to add Hadoop and Java environment variables.
- Reload the .bashrc file with source ~/.bashrc and verify Hadoop installation using hadoop version.
- Edit core-site.xml to set the default file system and temporary directory.
- Configure hdfs-site.xml to set replication factor, NameNode, and DataNode directories.
- Create mapred-site.xml and configure the MapReduce framework to use YARN.
- Edit yarn-site.xml to set up ResourceManager and NodeManager services.
- Create the necessary Hadoop directories for the NameNode, DataNode, and temporary storage.
- Format the Hadoop HDFS filesystem using hdfs namenode -format.
- Start the HDFS services with start-dfs.sh.
- Start YARN services with start-yarn.sh.
- Check running processes with jps to ensure services are active.
- Access Hadoop web interfaces: NameNode at http://localhost:9870/ and ResourceManager at http://localhost:8088/.

CODING: -

- sudo su
- sudo yum update -y
- sudo yum install java-11-openjdk-devel -y **or** sudo yum install java-1.8.0-amazon-corretto-devel -y
- java -version
- cd /usr/local/
- sudo wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
- sudo tar -xvzf hadoop-3.3.6.tar.gz
- sudo mv hadoop-3.3.6 hadoop

- sudo nano ~/.bashrc
    - # Hadoop variables
    - export HADOOP_HOME=/usr/local/hadoop
    - export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
    - # Java variables
    - export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:/bin/java::")
    - export PATH=$PATH:$JAVA_HOME/bin
- source ~/.bashrc
- hadoop version
- Configure core-site.xml
    - sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
    - <configuration>
    -   <property>
    -     <name>fs.defaultFS</name>
    -     <value>hdfs://localhost:9000</value>
    -   </property>
    -   <property>
    -     <name>hadoop.tmp.dir</name>
    -     <value>/usr/local/hadoop/tmp</value>
    -     <description>Temporary directory for Hadoop</description>
    -   </property>
    - </configuration>
- Configure hdfs-site.xml
    - sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
    - <configuration>
    -   <property>
    -     <name>dfs.replication</name>
    -     <value>1</value> <!-- Since this is a single-node setup -->
    -   </property>

  - `<property>`
  - `  <name>dfs.namenode.name.dir</name>`
  - `  <value>file:///usr/local/hadoop/hdfs/namenode</value>`
  - `</property>`
  - `<property>`
  - `  <name>dfs.datanode.data.dir</name>`
  - `  <value>file:///usr/local/hadoop/hdfs/datanode</value>`
  - `</property>`
  - `</configuration>`
- Create the mapred-site.xml[If the mapred-site.xml.template is not present]
  - sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml
  - `<configuration>`
  - `<property>`
  - `  <name>mapreduce.framework.name</name>`
  - `  <value>yarn</value>`
  - `</property>`
  - `</configuration>`
- Configure yarn-site.xml
  - sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
  - `<configuration>`
  - `<property>`
  - `  <name>yarn.nodemanager.aux-services</name>`
  - `  <value>mapreduce_shuffle</value>`
  - `</property>`
  - `<property>`
  - `  <name>yarn.resourcemanager.resource-tracker.address</name>`
  - `  <value>localhost:8025</value>`
  - `</property>`
  - `<property>`

-
  ```
      <name>yarn.resourcemanager.scheduler.address</name>
  ```

-
  ```
      <value>localhost:8030</value>
  ```

-
  ```
    </property>
  ```

-
  ```
    <property>
  ```

-
  ```
      <name>yarn.resourcemanager.address</name>
  ```

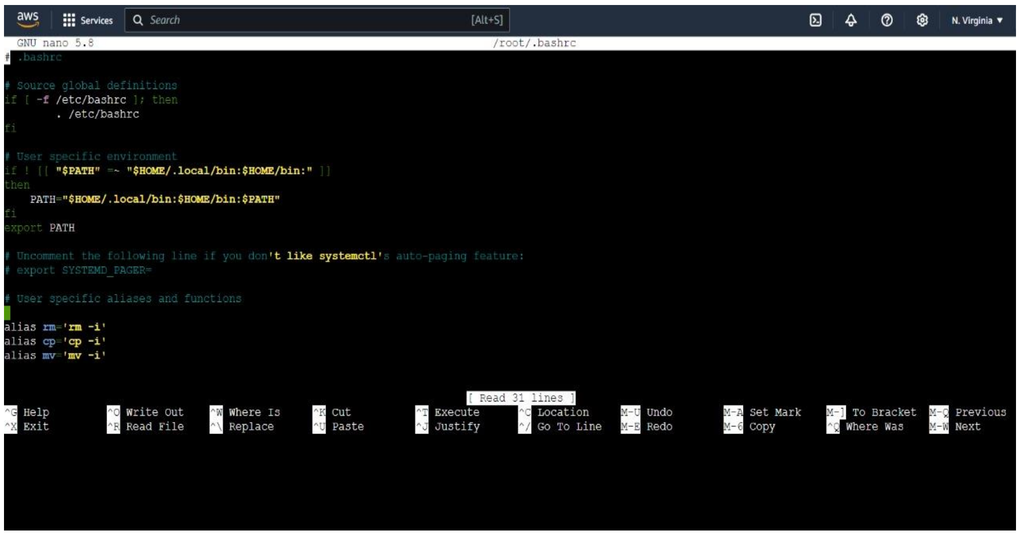
-
  ```
      <value>localhost:8050</value>
  ```

-
  ```
    </property>
  ```

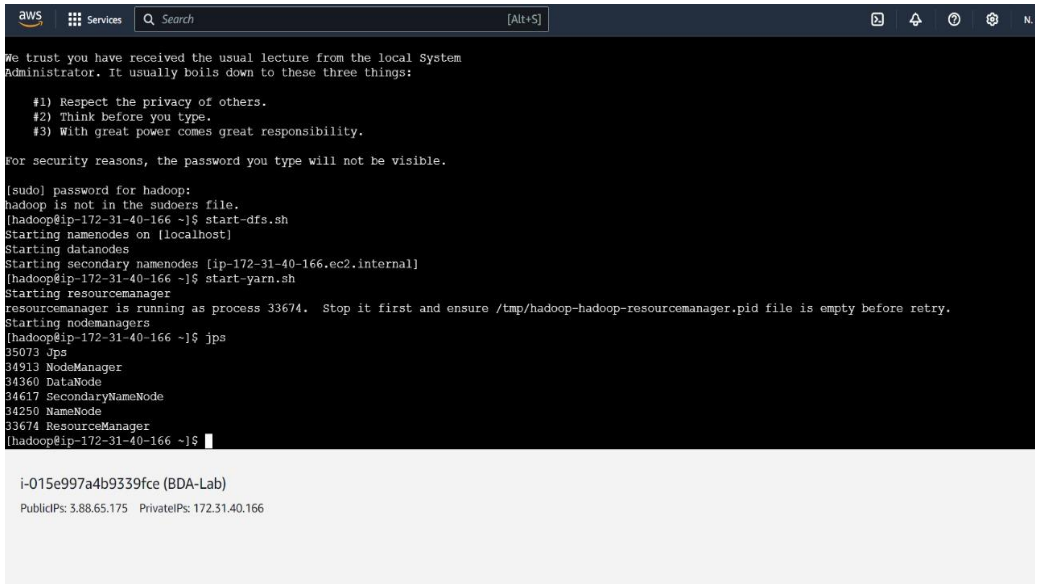
-
  ```
  </configuration>
  ```

- Set Up Hadoop Directories
  - sudo mkdir -p /usr/local/hadoop/hdfs/namenode
  - sudo mkdir -p /usr/local/hadoop/hdfs/datanode
  - sudo mkdir -p /usr/local/hadoop/tmp
- Format the HDFS Filesystem
  - hdfs namenode -format
- Start Hadoop Services(As Non - Root User)
  - start-dfs.sh
  - start-yarn.sh
  - Jps
- Access the Hadoop Web Interfaces
  - **NameNode**: http://localhost:9870/ (shows the HDFS overview)
  - **ResourceManager**: http://localhost:8088/ (shows the YARN overview)
  - **LocalHost - 127.0.0.1 or public DNS :- 3.117.182.16**

OUTPUT: -

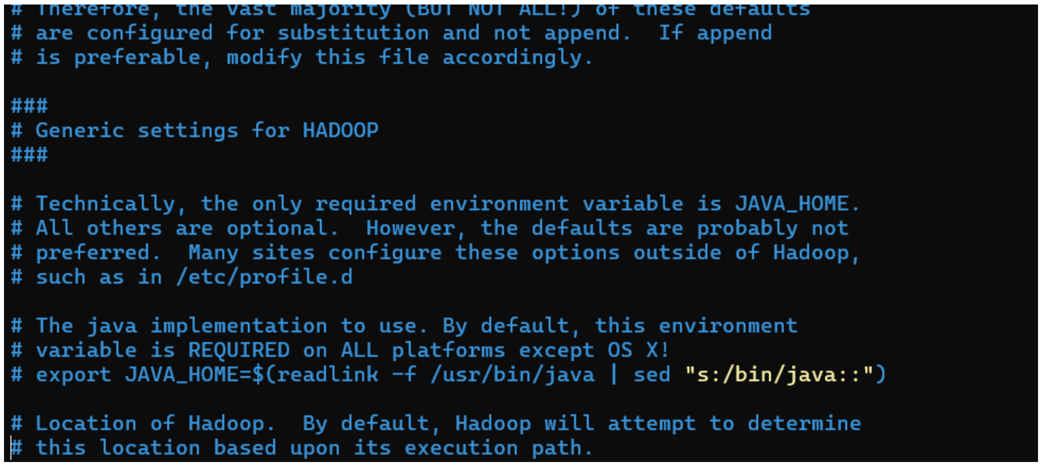


```
GNU nano 5.8                                    /root/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

                                        [ Read 31 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was  M-W Next
```



```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

For security reasons, the password you type will not be visible.

[sudo] password for hadoop:
hadoop is not in the sudoers file.
[hadoop@ip-172-31-40-166 ~]$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ip-172-31-40-166.ec2.internal]
[hadoop@ip-172-31-40-166 ~]$ start-yarn.sh
Starting resourcemanager
resourcemanager is running as process 33674.  Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
[hadoop@ip-172-31-40-166 ~]$ jps
35073 Jps
34913 NodeManager
34360 DataNode
34617 SecondaryNameNode
34250 NameNode
33674 ResourceManager
[hadoop@ip-172-31-40-166 ~]$
```

i-015e997a4b9339fce (BDA-Lab)
PublicIPs: 3.88.65.175   PrivateIPs: 172.31.40.166



```
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
# Generic settings for HADOOP
###

# 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=$(readlink -f /usr/bin/java | sed "s:/bin/java::")

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

PROCEDURE: -

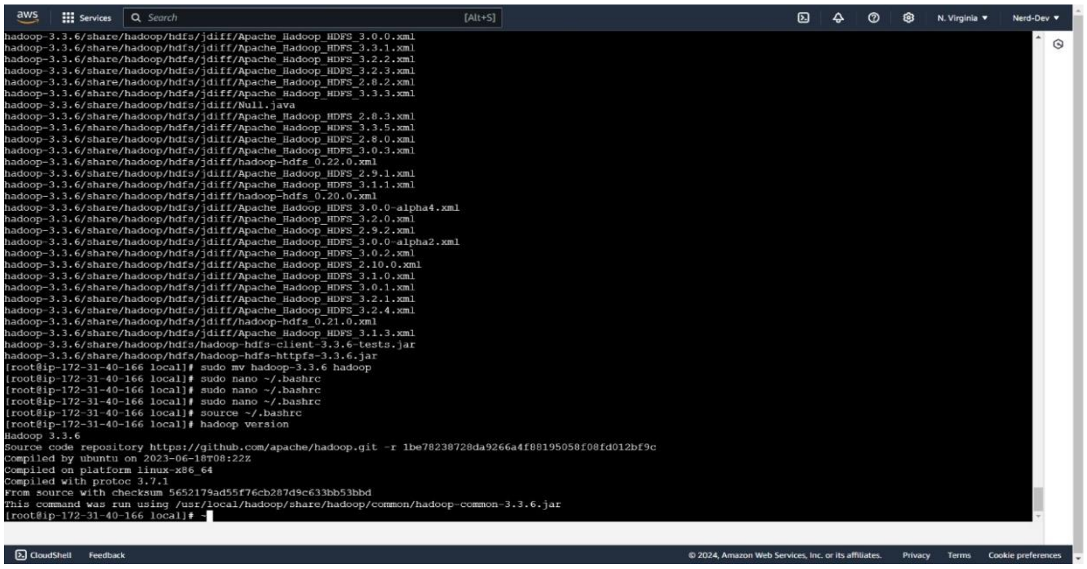
- Switch to superuser mode using sudo su.
- Adding Files to HDFS
    - To upload a local file to HDFS, use the hdfs dfs -put
- Listing Files and Directories
    - To list files and directories in a specific HDFS directory, use the hdfs dfs -ls command
- Creating Directories in HDFS
    - To create a new directory in HDFS, use the hdfs dfs -mkdir command
- Retrieving Files from HDFS
    - To download a file from HDFS to your local filesystem, use the hdfs dfs -get command
- Deleting Files from HDFS
    - To delete a file from HDFS, use the hdfs dfs -rm command
- Deleting Directories from HDFS
    - To delete a directory and its contents from HDFS, use the hdfs dfs -rm -r command
- Viewing File Contents in HDFS
    - To view the contents of a file in HDFS, use the hdfs dfs -cat command
- Checking Disk Usage of HDFS Directory
    - To check disk usage of files and directories in HDFS, use the hdfs dfs -du -h command
- Copying Files within HDFS
    - To copy a file or directory within HDFS, use the hdfs dfs -cp command
- Moving Files within HDFS
    - To move a file or directory within HDFS, use the hdfs dfs -mv command

CODING: -

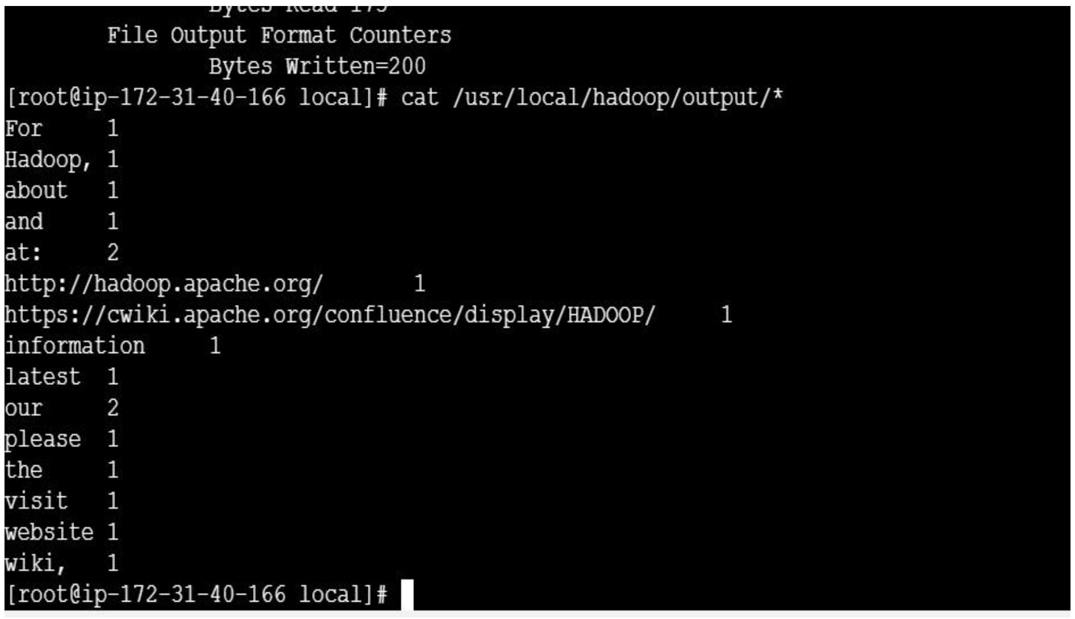
- sudo su
- hdfs dfs -put /local/path/to/file /hdfs/path/
- hdfs dfs -ls /hdfs/path/
- hdfs dfs -mkdir /hdfs/path/
- hdfs dfs -get /hdfs/path/to/file /local/path/
- hdfs dfs -rm /hdfs/path/to/file
- hdfs dfs -rm -r /hdfs/path/to/directory
- hdfs dfs -cat /hdfs/path/to/file
- hdfs dfs -du -h /hdfs/path/

- hdfs dfs -cp /hdfs/source/path /hdfs/destination/path
- hdfs dfs -mv /hdfs/source/path /hdfs/destination/path

OUTPUT: -



```
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Null.java
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.5.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.22.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.20.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha4.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.10.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.4.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.21.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.3.xml
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-client-3.3.6-tests.jar
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-httpfs-3.3.6.jar
[root@ip-172-31-40-166 local]# sudo mv hadoop-3.3.6 hadoop
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# source ~/.bashrc
[root@ip-172-31-40-166 local]# hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
[root@ip-172-31-40-166 local]# ~
```



```
        File Output Format Counters
                Bytes Written=200
[root@ip-172-31-40-166 local]# cat /usr/local/hadoop/output/*
For     1
Hadoop, 1
about   1
and     1
at:     2
http://hadoop.apache.org/       1
https://cwiki.apache.org/confluence/display/HADOOP/     1
information     1
latest  1
our     2
please  1
the     1
visit   1
website 1
wiki,   1
[root@ip-172-31-40-166 local]#
```

PROCEDURE: -

- Switch to superuser mode using sudo su.
- Input Format:
  - We'll have two input files representing matrices A and B.
  - Matrix A (m x n): Split by rows.
  - Matrix B (n x p): Split by columns.
- Map Step:
  - For each element in matrix A (i, k, A[i][k]), emit the product to the intermediate key (i, j) where j is the column index in matrix B.
  - For each element in matrix B (k, j, B[k][j]), emit the product to the intermediate key (i, j) for all rows i of matrix A.
- Reduce Step:
  - For each intermediate key (i, j), sum the products of the corresponding values to calculate the result matrix C (i, j).
- Output Format:
  - Output matrix C, where each line is of the form i, j, C[i][j].

CODING: -

- sudo su
- MatrixMapper.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MatrixMapper extends Mapper<Object, Text, Text, Text> {

    @Override
    public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
        String[] line = value.toString().split(",");
        String matrixName = line[0];  // A or B
        int i = Integer.parseInt(line[1]);
        int j = Integer.parseInt(line[2]);
        int valueOfElement = Integer.parseInt(line[3]);

        if (matrixName.equals("A")) {
            // Emit for all columns of B
            for (int k = 0; k < context.getConfiguration().getInt("p", 0); k++) {
                context.write(new Text(i + "," + k), new Text("A," + j + "," +
valueOfElement));
            }
        } else {
            // Emit for all rows of A
            for (int k = 0; k < context.getConfiguration().getInt("m", 0); k++) {
                context.write(new Text(k + "," + j), new Text("B," + i + "," +
valueOfElement));
            }
        }
    }
}
```

- MatrixReducer.java

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MatrixReducer extends Reducer<Text, Text, Text, IntWritable> {

    @Override
    public void reduce(Text key, Iterable<Text> values, Context context) throws
IOException, InterruptedException {
        Map<Integer, Integer> mapA = new HashMap<>();
        Map<Integer, Integer> mapB = new HashMap<>();

        for (Text val : values) {
            String[] parts = val.toString().split(",");
            if (parts[0].equals("A")) {
                mapA.put(Integer.parseInt(parts[1]), Integer.parseInt(parts[2]));
            } else {
                mapB.put(Integer.parseInt(parts[1]), Integer.parseInt(parts[2]));
            }
        }

        int result = 0;
        for (Integer k : mapA.keySet()) {
            if (mapB.containsKey(k)) {
                result += mapA.get(k) * mapB.get(k);
            }
        }

        context.write(key, new IntWritable(result));
    }
}
```

- MatrixMultiplication.java

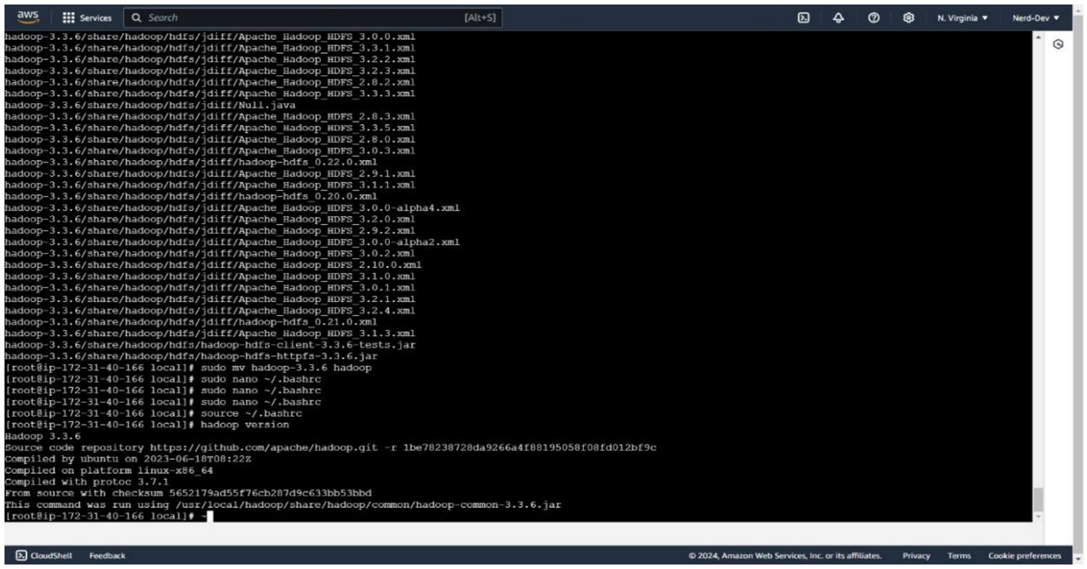
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

```java
public class MatrixMultiplication {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        // Dimensions of the matrices A (m x n) and B (n x p)
        conf.setInt("m", 3);  // Rows of A
        conf.setInt("n", 2);  // Columns of A and Rows of B
        conf.setInt("p", 3);  // Columns of B

        Job job = Job.getInstance(conf, "Matrix Multiplication");
        job.setJarByClass(MatrixMultiplication.class);
        job.setMapperClass(MatrixMapper.class);
        job.setReducerClass(MatrixReducer.class);

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

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

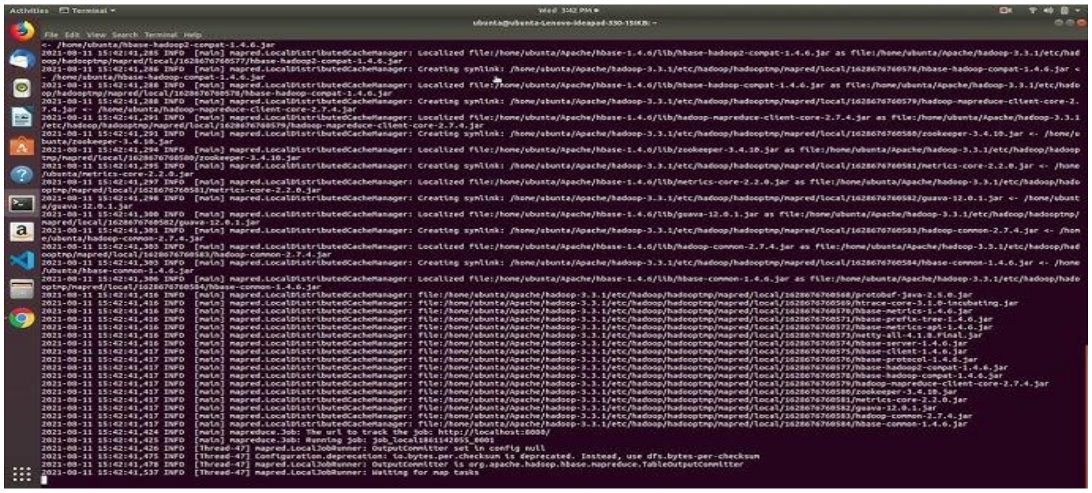
- Steps to Run the Code

```
hdfs dfs -put matrixA.txt /input/
hdfs dfs -put matrixB.txt /input/
hadoop com.sun.tools.javac.Main MatrixMultiplication.java
jar cf matrixmultiplication.jar MatrixMultiplication*.class
hadoop jar matrixmultiplication.jar MatrixMultiplication /input/ /output/
hdfs dfs -cat /output/part-r-00000
```

OUTPUT: -



```
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Null.java
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.3.5.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.8.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.3.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.22.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.20.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha4.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.9.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.0-alpha2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.2.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_2.10.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.0.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.1.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.2.4.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/hadoop-hdfs_0.21.0.xml
hadoop-3.3.6/share/hadoop/hdfs/jdiff/Apache_Hadoop_HDFS_3.1.3.xml
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-client-3.3.6-tests.jar
hadoop-3.3.6/share/hadoop/hdfs/hadoop-hdfs-httpfs-3.3.6.jar
[root@ip-172-31-40-166 local]# sudo mv hadoop-3.3.6 hadoop
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# sudo nano ~/.bashrc
[root@ip-172-31-40-166 local]# source ~/.bashrc
[root@ip-172-31-40-166 local]# hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
[root@ip-172-31-40-166 local]# ~
```



```
<- /home/ubunta/hbase-hadoop2-compat-1.4.6.jar
2021-08-11 15:42:41,285 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/hbase-hadoop2-compat-1.4.6.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/had
oop/hadooptmp/mapred/local/1628676760577/hbase-hadoop2-compat-1.4.6.jar
2021-08-11 15:42:41,286 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760578/hbase-hadoop-compat-1.4.6.jar <
- /home/ubunta/hbase-hadoop-compat-1.4.6.jar
2021-08-11 15:42:41,288 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/hbase-hadoop-compat-1.4.6.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hado
op/hadooptmp/mapred/local/1628676760578/hbase-hadoop-compat-1.4.6.jar
2021-08-11 15:42:41,288 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760579/hadoop-mapreduce-client-core-2.
7.4.jar <- /home/ubunta/hadoop-mapreduce-client-core-2.7.4.jar
2021-08-11 15:42:41,291 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/hadoop-mapreduce-client-core-2.7.4.jar as file:/home/ubunta/Apache/hadoop-3.3.1
/etc/hadoop/hadooptmp/mapred/local/1628676760579/hadoop-mapreduce-client-core-2.7.4.jar
2021-08-11 15:42:41,291 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760580/zookeeper-3.4.10.jar <- /home/u
bunta/zookeeper-3.4.10.jar
2021-08-11 15:42:41,294 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/zookeeper-3.4.10.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadoop
tmp/mapred/local/1628676760580/zookeeper-3.4.10.jar
2021-08-11 15:42:41,295 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760581/metrics-core-2.2.0.jar <- /home
/ubunta/metrics-core-2.2.0.jar
2021-08-11 15:42:41,297 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/metrics-core-2.2.0.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hado
optmp/mapred/local/1628676760581/metrics-core-2.2.0.jar
2021-08-11 15:42:41,298 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760582/guava-12.0.1.jar <- /home/ubunt
a/guava-12.0.1.jar
2021-08-11 15:42:41,300 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/guava-12.0.1.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/
mapred/local/1628676760582/guava-12.0.1.jar
2021-08-11 15:42:41,301 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760583/hadoop-common-2.7.4.jar <- /hom
e/ubunta/hadoop-common-2.7.4.jar
2021-08-11 15:42:41,303 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/hadoop-common-2.7.4.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/had
ooptmp/mapred/local/1628676760583/hadoop-common-2.7.4.jar
2021-08-11 15:42:41,303 INFO  [main] mapred.LocalDistributedCacheManager: Creating symlink: /home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760584/hbase-common-1.4.6.jar <- /home
/ubunta/hbase-common-1.4.6.jar
2021-08-11 15:42:41,306 INFO  [main] mapred.LocalDistributedCacheManager: Localized file:/home/ubunta/Apache/hbase-1.4.6/lib/hbase-common-1.4.6.jar as file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hado
optmp/mapred/local/1628676760584/hbase-common-1.4.6.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760568/protobuf-java-2.5.0.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760569/htrace-core-3.1.0-incubating.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760570/hbase-metrics-1.4.6.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760571/hbase-prefix-tree-1.4.6.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760572/hbase-metrics-api-1.4.6.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760573/netty-all-4.1.8.Final.jar
2021-08-11 15:42:41,416 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760574/hbase-server-1.4.6.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760575/hbase-client-1.4.6.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760576/hbase-protocol-1.4.6.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760577/hbase-hadoop2-compat-1.4.6.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760578/hbase-hadoop-compat-1.4.6.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760579/hadoop-mapreduce-client-core-2.7.4.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760580/zookeeper-3.4.10.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760581/metrics-core-2.2.0.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760582/guava-12.0.1.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760583/hadoop-common-2.7.4.jar
2021-08-11 15:42:41,417 INFO  [main] mapred.LocalDistributedCacheManager: file:/home/ubunta/Apache/hadoop-3.3.1/etc/hadoop/hadooptmp/mapred/local/1628676760584/hbase-common-1.4.6.jar
2021-08-11 15:42:41,424 INFO  [main] mapreduce.Job: The url to track the job: http://localhost:8080/
2021-08-11 15:42:41,425 INFO  [main] mapreduce.Job: Running job: job_local1861142055_0001
2021-08-11 15:42:41,426 INFO  [Thread-47] mapred.LocalJobRunner: OutputCommitter set in config null
2021-08-11 15:42:41,475 INFO  [Thread-47] Configuration.deprecation: io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2021-08-11 15:42:41,478 INFO  [Thread-47] mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.hbase.mapreduce.TableOutputCommitter
2021-08-11 15:42:41,537 INFO  [Thread-47] mapred.LocalJobRunner: Waiting for map tasks
```

Activities
Terminal
Wed 3:42 PM
ubunta@ubunta-Lenovo-ideapad-330-15IKB: ~
File Edit View Search Terminal Help

PROCEDURE: -

- Switch to superuser mode using sudo su.
- Create a text file on your local filesystem that contains some sample data (e.g., input.txt).
- Put the file into HDFS (Hadoop Distributed File System)
- The mapper reads the input line by line, splits each line into words, and outputs each word as a key with a value of 1.
- The reducer sums the counts for each word emitted by the mapper and outputs the word along with its total count
- Compile the Java Code
- Run the MapReduce Job
- Once the job is complete, you can check the output by viewing the result file in HDFS

CODING: -

- sudo su
- WordCountMapper.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String[] words = value.toString().split("\\s+");
        for (String str : words) {
```

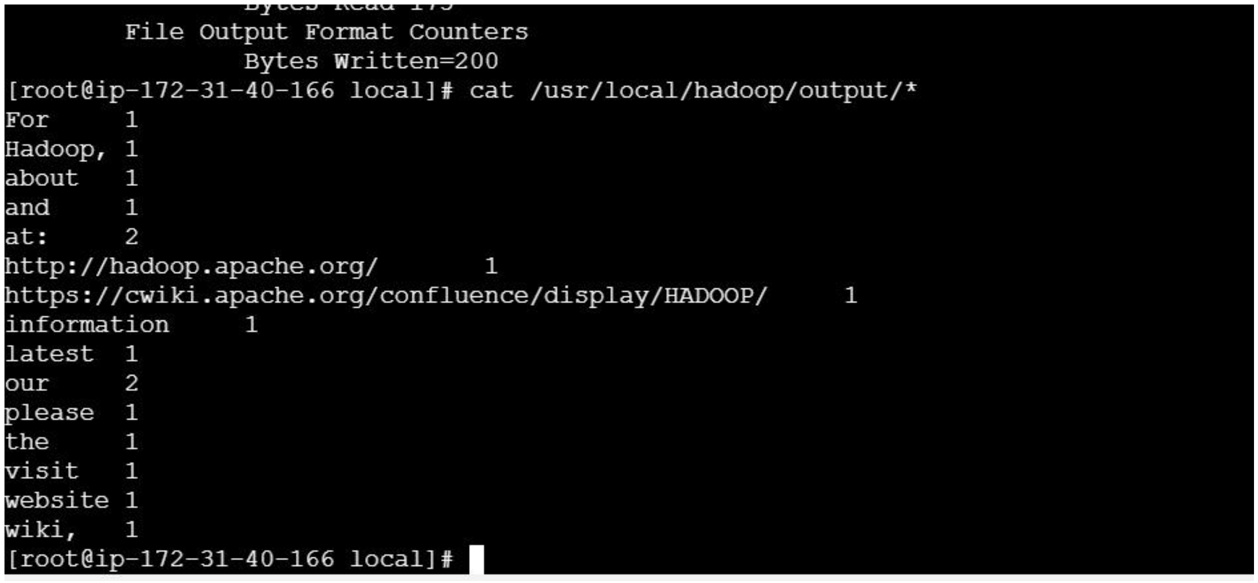
-
  ```
              word.set(str.replaceAll("[^a-zA-Z]", "").toLowerCase());  // Normalize word
              if (!word.toString().isEmpty()) {
                  context.write(word, one);
              }
          }
      }
  }
  ```
- WordCountReducer.java
  ```
  import java.io.IOException;
  import org.apache.hadoop.io.IntWritable;
  import org.apache.hadoop.io.Text;
  import org.apache.hadoop.mapreduce.Reducer;

  public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

      @Override
      protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
          int sum = 0;
          for (IntWritable val : values) {
              sum += val.get();
          }
          context.write(key, new IntWritable(sum));
      }
  }
  ```

- WordCount.java

```java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Word Count");

        job.setJarByClass(WordCount.class);
        job.setMapperClass(WordCountMapper.class);
        job.setReducerClass(WordCountReducer.class);

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

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

- Steps to Run the Code
    - hadoop com.sun.tools.javac.Main WordCount.java
    - jar cf wordcount.jar WordCount*.class
    - hadoop jar wordcount.jar WordCount /input /output
    - hdfs dfs -cat /output/part-r-00000

OUTPUT: -



```
        File Output Format Counters
                Bytes Written=200
[root@ip-172-31-40-166 local]# cat /usr/local/hadoop/output/*
For     1
Hadoop, 1
about   1
and     1
at:     2
http://hadoop.apache.org/       1
https://cwiki.apache.org/confluence/display/HADOOP/     1
information     1
latest  1
our     2
please  1
the     1
visit   1
website 1
wiki,   1
[root@ip-172-31-40-166 local]# 
```

PROCEDURE: -

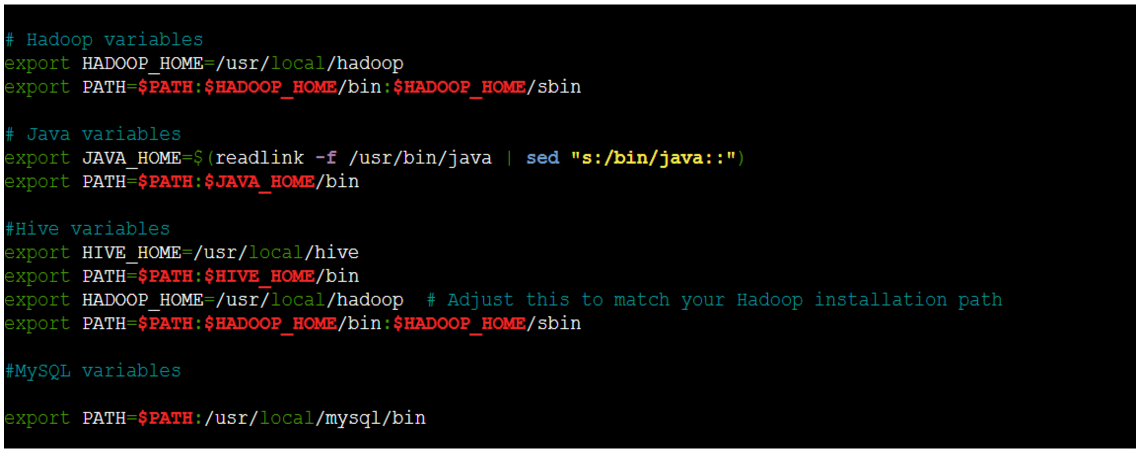
- Switch to superuser mode using sudo su.
- Update all system packages to ensure you have the latest versions.
- Install wget to download files from the internet.
- Use wget to download the Apache Hive binary tarball from the official website
- Extract the Hive binary tarball using the tar command.
- Move the extracted Hive directory to /usr/local/hive for proper management.
- Open the .bashrc file to set the necessary environment variables for Hive.
- After editing the .bashrc file, apply the changes by sourcing it.
- You can verify that Hive has been successfully installed by running the hive command.

CODING: -

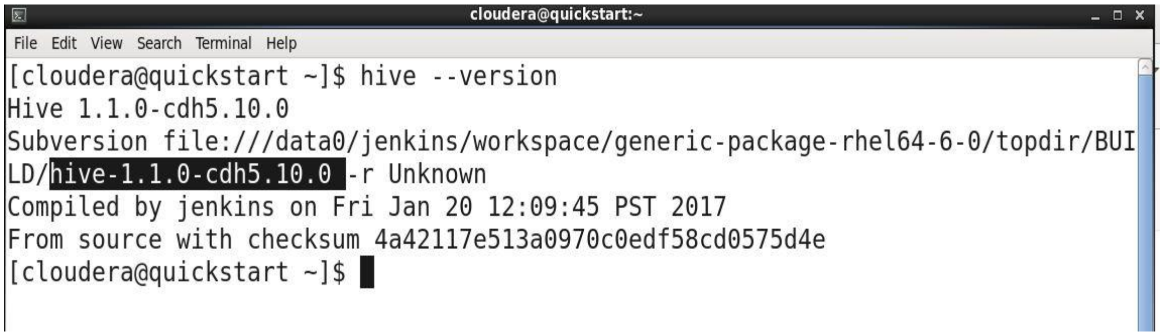
- sudo su

- **Update the system**
  - sudo yum update -y
- **Install wget**
  - sudo yum install wget -y
- **Download Apache Hive**
  - wget https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz
- **Extract the Hive tarball**
  - tar -xzvf apache-hive-3.1.3-bin.tar.gz
- **Move to root directory**
  - sudo mv apache-hive-3.1.3-bin /usr/local/hive
- **Configure Hive Environment Variables**
  - nano ~/.bashrc
  - Add the following in that file as #Hive Variables
    - export HIVE_HOME=/usr/local/hive
    - export PATH=$PATH:$HIVE_HOME/bin
    - export HADOOP_HOME=/usr/local/hadoop   # Adjust to your Hadoop installation path
    - export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
  - source ~/.bashrc

OUTPUT: -



```
# Hadoop variables
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

# Java variables
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:/bin/java::")
export PATH=$PATH:$JAVA_HOME/bin

#Hive variables
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_HOME=/usr/local/hadoop  # Adjust this to match your Hadoop installation path
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

#MySQL variables

export PATH=$PATH:/usr/local/mysql/bin
```



```
[cloudera@quickstart ~]$ hive --version
Hive 1.1.0-cdh5.10.0
Subversion file:///data0/jenkins/workspace/generic-package-rhel64-6-0/topdir/BUI
LD/hive-1.1.0-cdh5.10.0 -r Unknown
Compiled by jenkins on Fri Jan 20 12:09:45 PST 2017
From source with checksum 4a42117e513a0970c0edf58cd0575d4e
[cloudera@quickstart ~]$
```

PROCEDURE: -

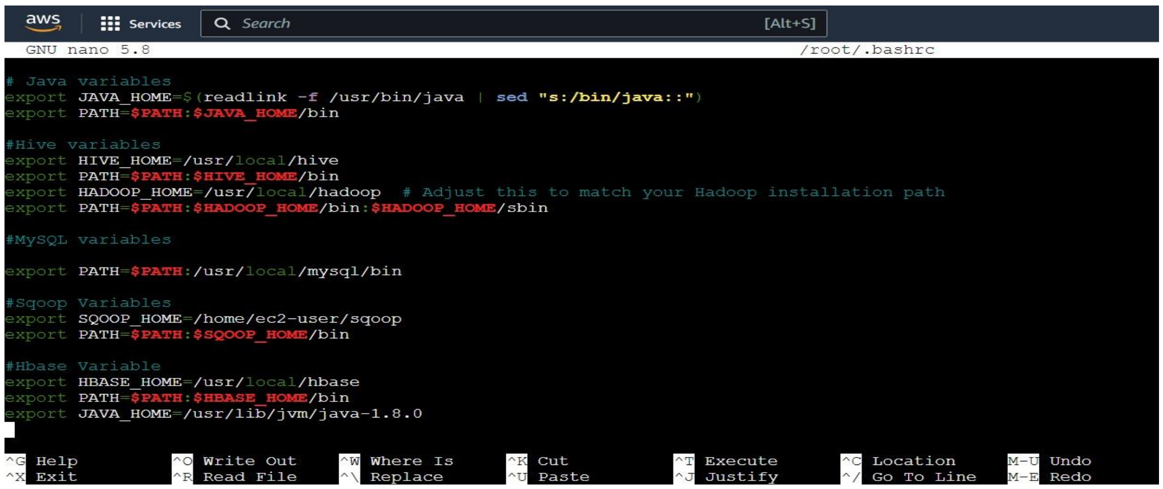
- Switch to superuser mode using sudo su.
- Update all system packages to ensure you have the latest versions.
- Install wget to download files from the internet.
- Use wget to download the HBase binary tarball from the official website.
- Extract the HBase binary tarball using the tar command.
- Move the extracted HBase directory to /usr/local/hbase for proper management.
- Open the .bashrc file to set the necessary environment variables for HBase.
- After editing the .bashrc file, apply the changes by sourcing it.
- Navigate to the HBase configuration directory.
- Open and edit the hbase-env.sh file to configure the environment variables for HBase.
- Open the hbase-site.xml file to configure HBase's storage mode.
- Finally, start the HBase services using the following command.
- You can verify if HBase has started correctly by running the HBase shell

CODING: -

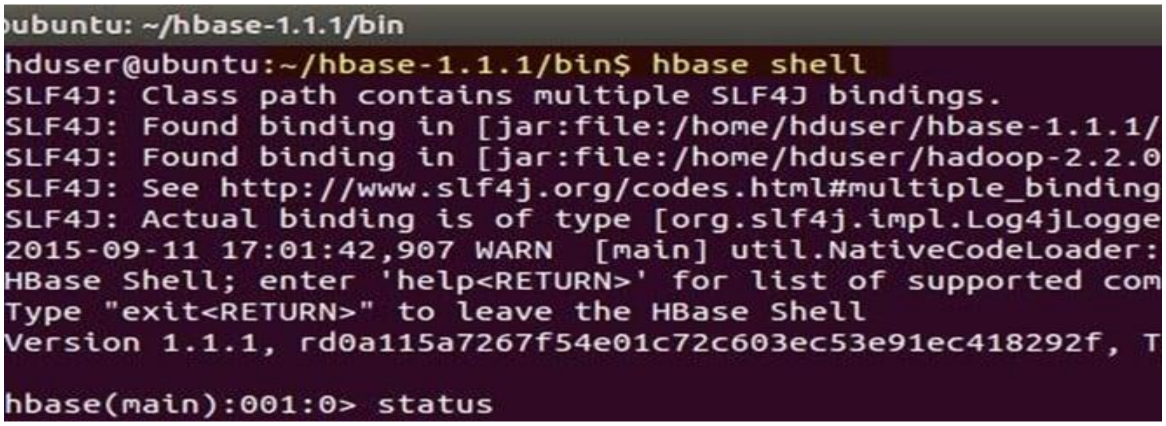
- sudo su
- **Update the system**
  - sudo yum update -y
- **Install wget**
  - sudo yum install wget -y
- **Download HBase**
  - wget https://downloads.apache.org/hbase/2.4.15/hbase-2.4.15-bin.tar.gz
- **Extract the HBase tarball**
  - tar -xzvf hbase-2.4.15-bin.tar.gz
- **Move to root directory**
  - sudo mv hbase-2.4.15 /usr/local/hbase
- **Configure Hbase Environment Variables**
  - nano ~/.bashrc
  - Add the following in that file as #HBase Variables
    - export HBASE_HOME=/usr/local/hbase
    - export PATH=$PATH:$HBASE_HOME/bin
    - export JAVA_HOME=/usr/lib/jvm/java-1.8.0source ~/.bashrc
  - source ~/.bashrc
- **Configure HBase**
  - cd /usr/local/hbase/conf
  - sudo nano hbase-env.sh
  - export JAVA_HOME=/usr/lib/jvm/java-1.8.0
- **Configure HBase storage mode**
  - sudo nano hbase-site.xml
    - <configuration>
    -   <property>
    -     <name>hbase.rootdir</name>

    - <value>file:///usr/local/hbase/data</value>
    - </property>
    - <property>
    - <name>hbase.zookeeper.property.dataDir</name>
    - <value>/usr/local/hbase/zookeeper</value>
    - </property>
    - </configuration>
- start-hbase.sh

OUTPUT: -



```
GNU nano 5.8                                                              /root/.bashrc

# Java variables
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:/bin/java::")
export PATH=$PATH:$JAVA_HOME/bin

#Hive variables
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_HOME=/usr/local/hadoop  # Adjust this to match your Hadoop installation path
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

#MySQL variables

export PATH=$PATH:/usr/local/mysql/bin

#Sqoop Variables
export SQOOP_HOME=/home/ec2-user/sqoop
export PATH=$PATH:$SQOOP_HOME/bin

#Hbase Variable
export HBASE_HOME=/usr/local/hbase
export PATH=$PATH:$HBASE_HOME/bin
export JAVA_HOME=/usr/lib/jvm/java-1.8.0

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



```
ubuntu: ~/hbase-1.1.1/bin
hduser@ubuntu:~/hbase-1.1.1/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hbase-1.1.1/
SLF4J: Found binding in [jar:file:/home/hduser/hadoop-2.2.0
SLF4J: See http://www.slf4j.org/codes.html#multiple_binding
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLogge
2015-09-11 17:01:42,907 WARN  [main] util.NativeCodeLoader:
HBase Shell; enter 'help<RETURN>' for list of supported com
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.1, rd0a115a7267f54e01c72c603ec53e91ec418292f, T

hbase(main):001:0> status
```

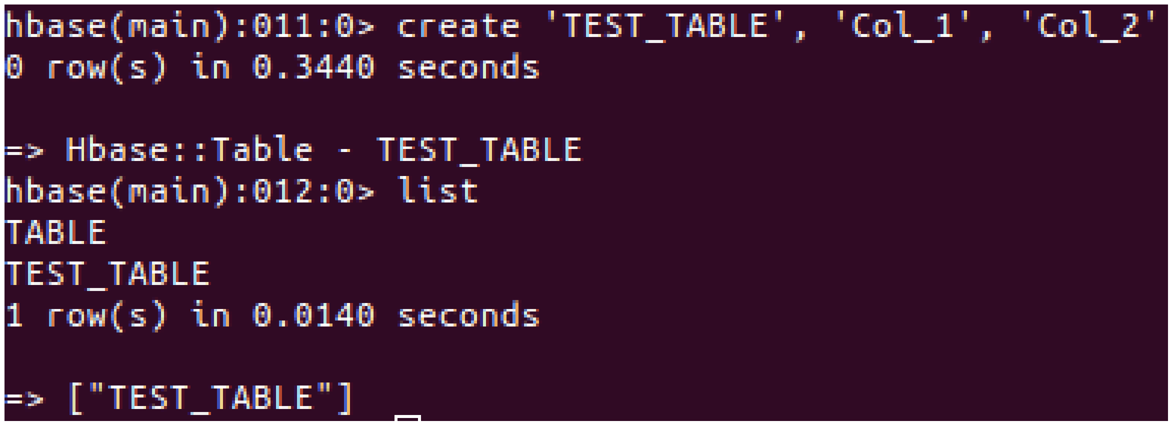
PROCEDURE: -

- Switch to superuser mode using sudo su.
- Open the HBase shell by executing the following command
- To create a table, use the create command
- To insert data into the table, use the put command.
- To retrieve data from the table, use the get command
- To scan data from the table, use the scan command
- To update a value, you can use the put command
- To delete a specific cell, use the delete command

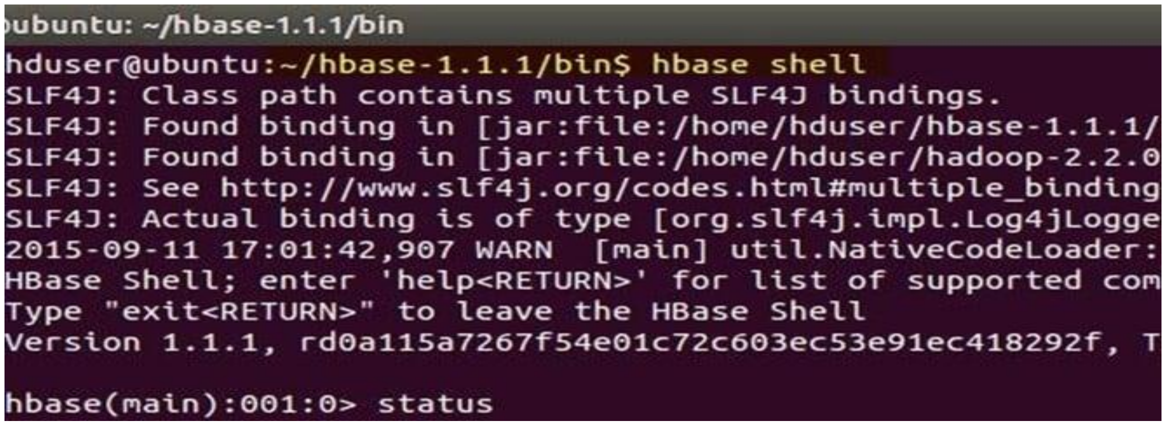
CODING: -

- sudo su
- # Start HBase shell
- hbase shell
- # Create a table
- create 'my_table', 'my_column_family'
- # Insert data (Create)
- put 'my_table', 'row1', 'my_column_family:name', 'Alice'
- put 'my_table', 'row1', 'my_column_family:age', '30'
- put 'my_table', 'row2', 'my_column_family:name', 'Bob'
- put 'my_table', 'row2', 'my_column_family:age', '25'
- # Read data (Retrieve)
- get 'my_table', 'row1'
- scan 'my_table'
- # Update data
- put 'my_table', 'row1', 'my_column_family:age', '31'
- # Delete data
- delete 'my_table', 'row1', 'my_column_family:age'
- deleteall 'my_table', 'row1' # Deletes entire row1
- # Drop the table (optional)
- drop 'my_table'

OUTPUT: -



```
hbase(main):011:0> create 'TEST_TABLE', 'Col_1', 'Col_2'
0 row(s) in 0.3440 seconds

=> Hbase::Table - TEST_TABLE
hbase(main):012:0> list
TABLE
TEST_TABLE
1 row(s) in 0.0140 seconds

=> ["TEST_TABLE"]
```



```
ubuntu: ~/hbase-1.1.1/bin
hduser@ubuntu:~/hbase-1.1.1/bin$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hbase-1.1.1/
SLF4J: Found binding in [jar:file:/home/hduser/hadoop-2.2.0
SLF4J: See http://www.slf4j.org/codes.html#multiple_binding
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLogge
2015-09-11 17:01:42,907 WARN  [main] util.NativeCodeLoader:
HBase Shell; enter 'help<RETURN>' for list of supported com
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.1, rd0a115a7267f54e01c72c603ec53e91ec418292f, T

hbase(main):001:0> status
```

PROCEDURE: -

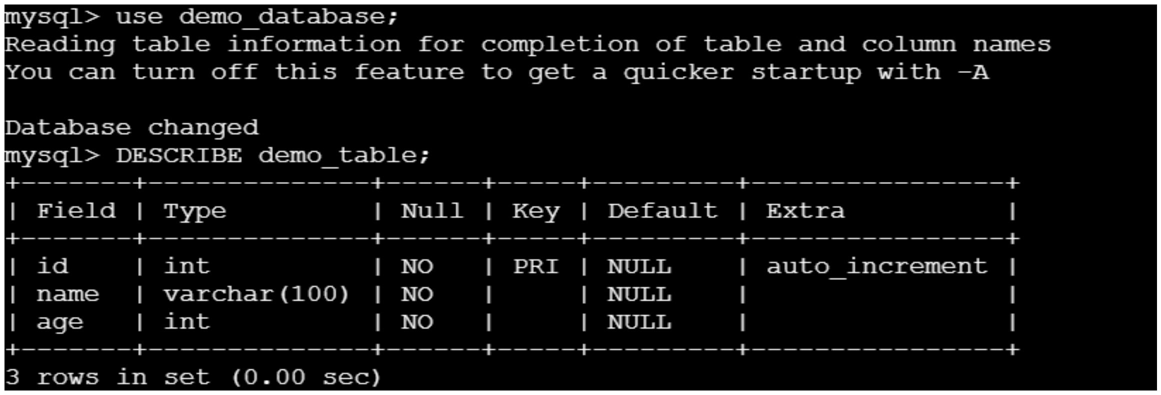
- Switch to superuser mode using sudo su.
- Import Data from MySQL to Hive
  - Check Sqoop Installation
  - Check MySQL Schema and Table Structure
    - Log in to MySQL to check the schema and the structure of the table you want to import
    - Describe the table to check its structure
  - Import MySQL Data into Hive using Sqoop
  - Verify the Hive Table
  - Describe the Hive table
- Export Data from Hive to MySQL
  - Check Hive Table Structure
  - Export Hive Data to MySQL with the use of sqoop commands
  - Check the Exported Data in MySQL

CODING: -

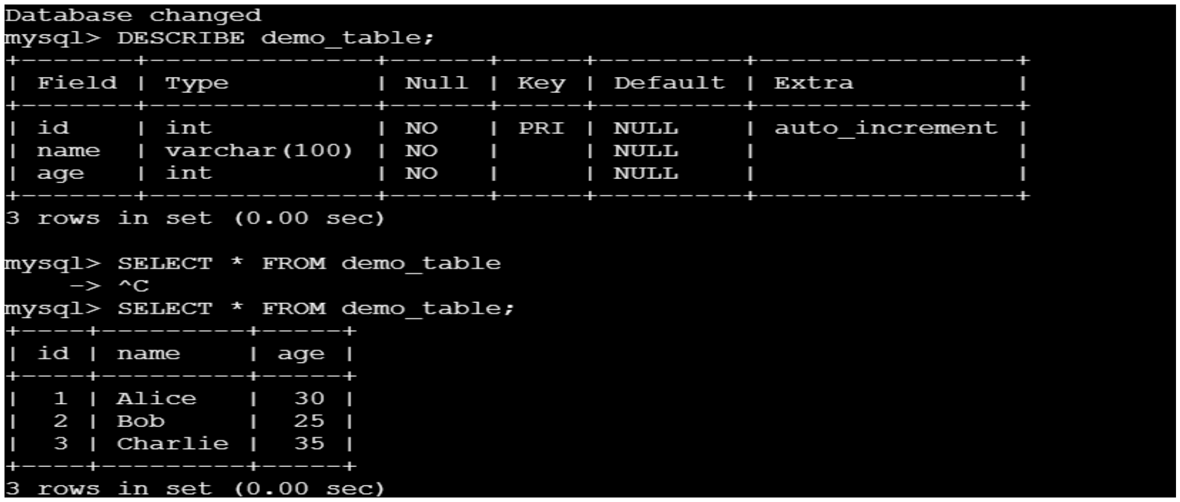
- sudo su
- # Part 1: Import Data from MySQL to Hive
  - # Check Sqoop Installation
  - sqoop version
  - # Check MySQL Schema and Table Structure
  - mysql -u your_user -p
  - DESCRIBE your_database.your_table;
  - # Import MySQL Data into Hive using Sqoop
  - sqoop import \
  - --connect jdbc:mysql://your-mysql-server-ip:3306/your_database \
  - --username your_user \
  - --password your_password \
  - --table your_table \
  - --hive-import \
  - --hive-table your_hive_database.your_hive_table \
  - --m 1

  - # Verify the Hive Table
  - hive
  - DESCRIBE your_hive_database.your_hive_table;
  - SELECT * FROM your_hive_database.your_hive_table LIMIT 10;
- # Part 2: Export Data from Hive to MySQL
  - # Check Hive Table Structure
  - DESCRIBE your_hive_database.your_hive_table;
  - # Export Hive Data to MySQL
  - sqoop export \
  - --connect jdbc:mysql://your-mysql-server-ip:3306/your_database \
  - --username your_user \
  - --password your_password \
  - --table your_table \
  - --export-dir /user/hive/warehouse/your_hive_table \
  - --m 1
  - # Check the Exported Data in MySQL
  - SELECT * FROM your_database.your_table LIMIT 10;

OUTPUT: -



```
mysql> use demo_database;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> DESCRIBE demo_table;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int          | NO   | PRI | NULL    | auto_increment |
| name  | varchar(100) | NO   |     | NULL    |                |
| age   | int          | NO   |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```



```
Database changed
mysql> DESCRIBE demo_table;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int          | NO   | PRI | NULL    | auto_increment |
| name  | varchar(100) | NO   |     | NULL    |                |
| age   | int          | NO   |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM demo_table
    -> ^C
mysql> SELECT * FROM demo_table;
+----+---------+-----+
| id | name    | age |
+----+---------+-----+
|  1 | Alice   |  30 |
|  2 | Bob     |  25 |
|  3 | Charlie |  35 |
+----+---------+-----+
3 rows in set (0.00 sec)
```