PRACTICAL REPORT
ON
PSCSP3042: BIG DATA ANALYTICS

SUBMITTED BY
MANEESH NARESH MASHELKAR

ROLL NO: 11

SUBMITTED TO
Mr. SUJAL SHAH

MSc. (COMPUTER SCIENCE) SEM - III
2022 – 2023

CONDUCTED AT
CHIKITSAK SAMUHA’S
S. S. & L.S. PATKAR COLLEGE OF ARTS & SCIENCE
AND
V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS
An Autonomous college,
Affiliated to University of Mumbai
GOREGAON (W). MUMBAI -400062

CHIKITSAK SAMUHA'S

SIR SITARAM & LADY SHANTABAI PATKAR COLLEGE OF ARTS & SCIENCE & V.P. VARDE COLLEGE OF COMMERCE & ECONOMICS

GOREGAON (WEST), MUMBAI - 400 104.

An Autonomous College, University of Mumbai

CERTIFICATE

Certified that such of the experiments as have been duly signed were performed by Mr./Miss ____________________

Roll No. ____________ of ____________ class ____________

Division ____________ in the ____________ Laboratory

of this college during the year ____________

Professor-in-Charge **Examiner** **Co-ordinator**

Date: ____________

____________ *Department*

INDEX

Practical No: 1

Aim: Installing and setting environment variables for Working with Apache Hadoop.

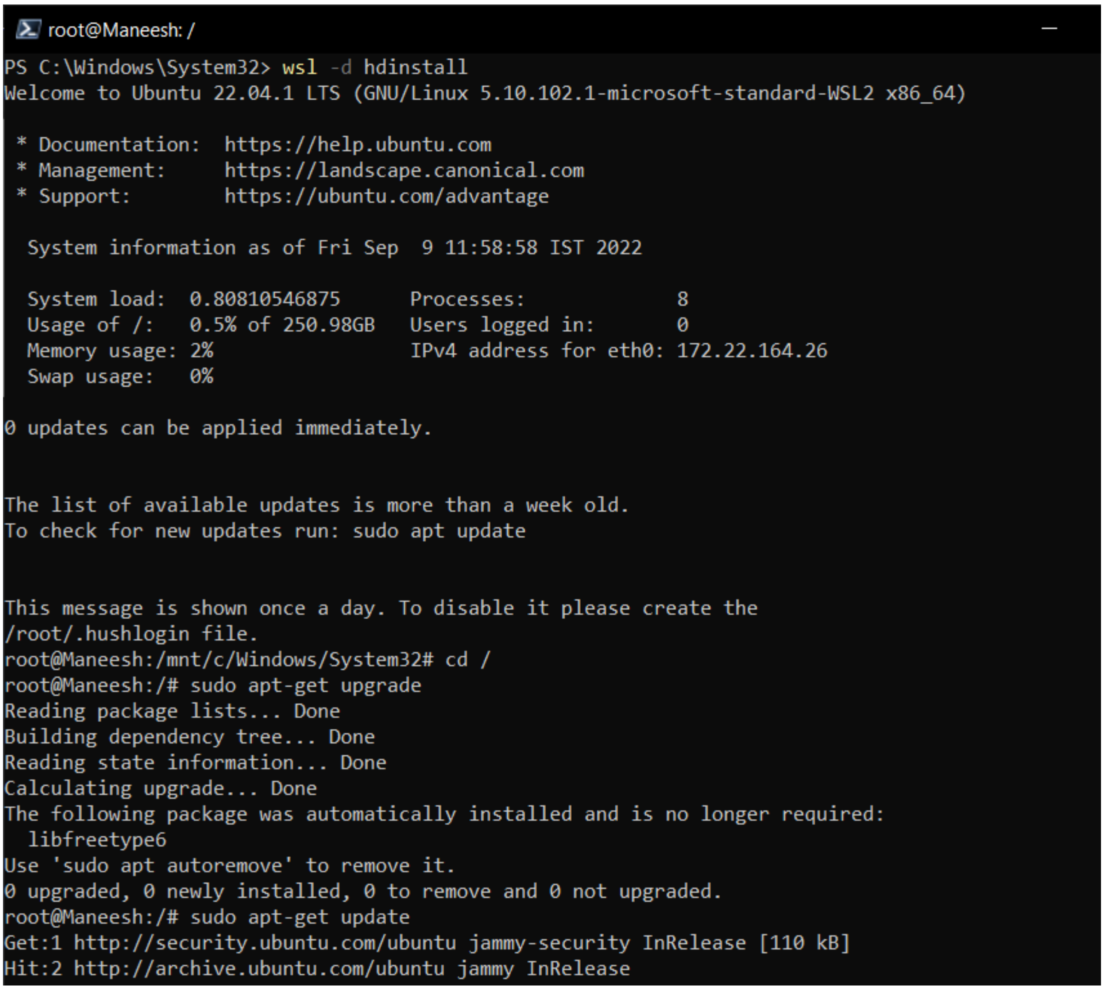
Theory:

Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. Hadoop was originally designed for computer clusters built from commodity hardware, which is still the common use. It has since also found use on clusters of higher-end hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

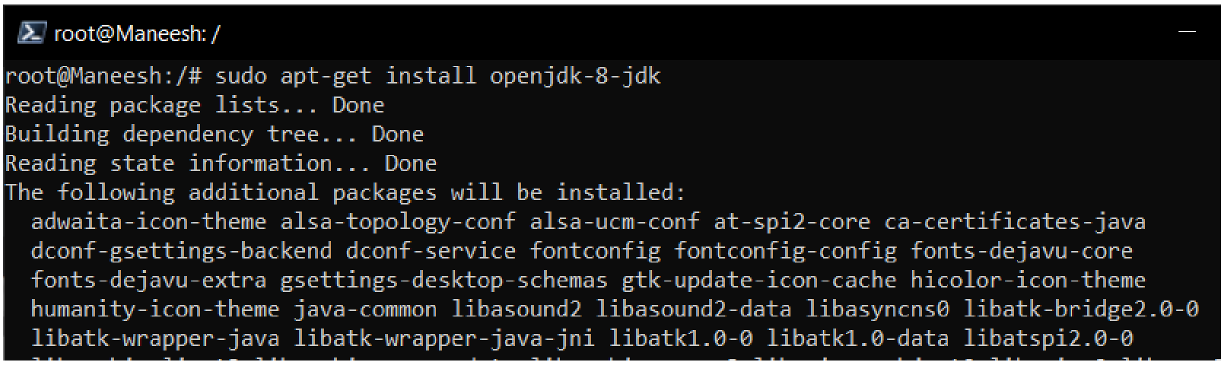
Steps:

1. First you need to perform some prerequisites
   - ➢ sudo apt-get upgrade
   - ➢ sudo apt-get update



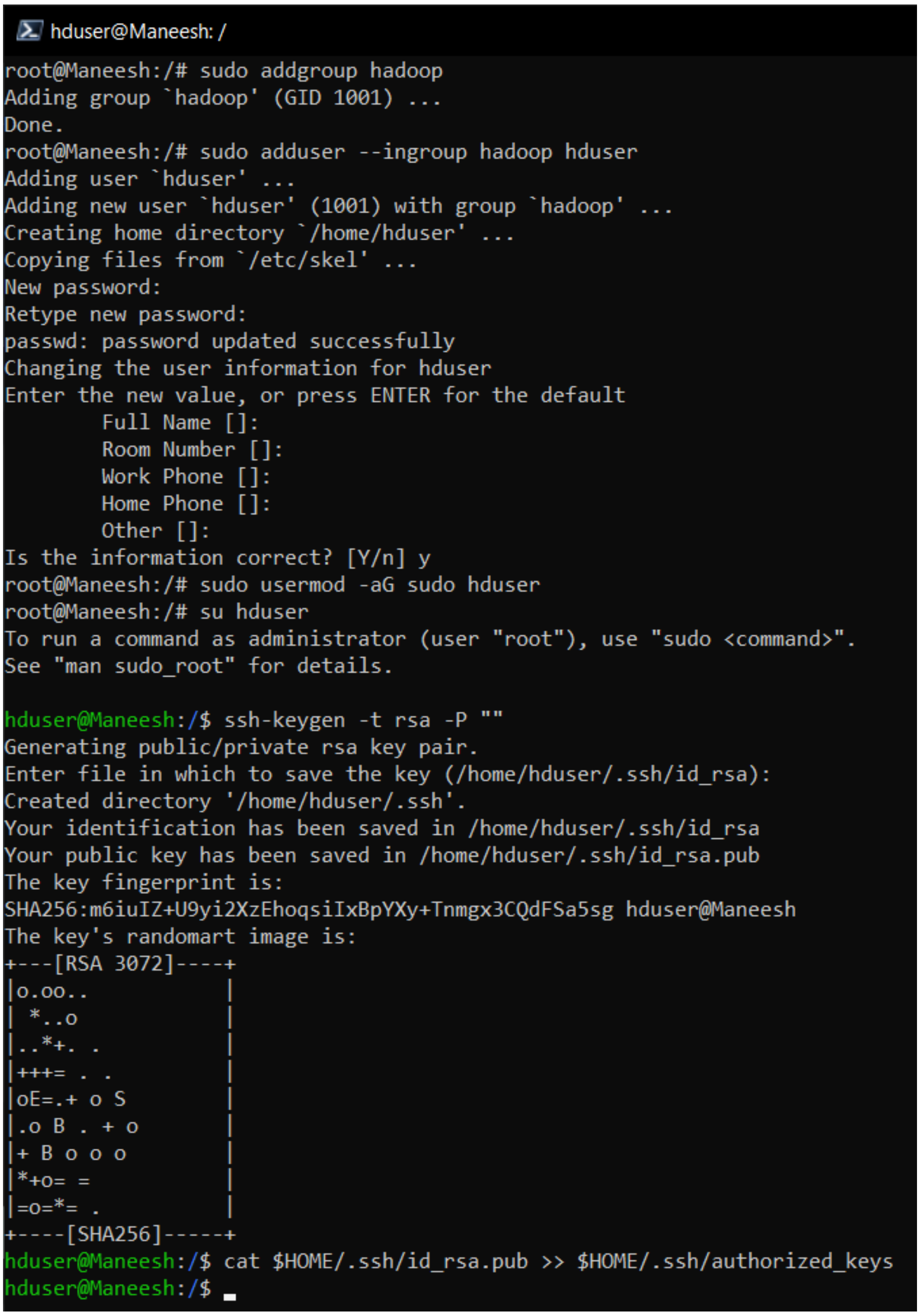
```
PS C:\Windows\System32> wsl -d hdinstall
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Sep  9 11:58:58 IST 2022

  System load:  0.80810546875      Processes:             8
  Usage of /:   0.5% of 250.98GB   Users logged in:       0
  Memory usage: 2%                 IPv4 address for eth0: 172.22.164.26
  Swap usage:   0%

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Maneesh:/mnt/c/Windows/System32# cd /
root@Maneesh:/# sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Maneesh:/# sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
```

- ➢ sudo apt-get install openjdk-8-jdk



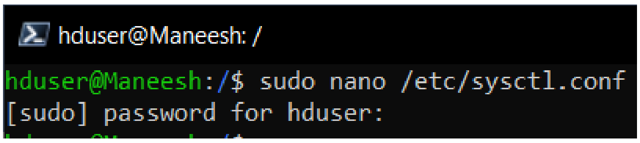
```
root@Maneesh:/# sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java
  dconf-gsettings-backend dconf-service fontconfig fontconfig-config fonts-dejavu-core
  fonts-dejavu-extra gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme java-common libasound2 libasound2-data libasyncns0 libatk-bridge2.0-0
  libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0
```

- ➢ sudo addgroup hadoop
- ➢ sudo adduser --ingroup hadoop hduser
- ➢ sudo usermod -aG sudo hduser
- ➢ su hduser
- ➢ ssh-keygen -t rsa -P ""
- ➢ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys



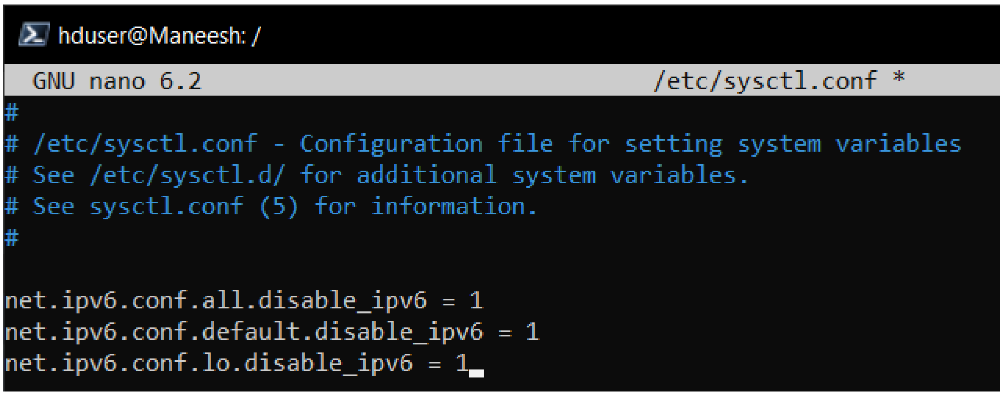
```
root@Maneesh:/# sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
root@Maneesh:/# sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@Maneesh:/# sudo usermod -aG sudo hduser
root@Maneesh:/# su hduser
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hduser@Maneesh:/$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:m6iuIZ+U9yi2XzEhoqsiIxBpYXy+Tnmgx3CQdFSa5sg hduser@Maneesh
The key's randomart image is:
+---[RSA 3072]----+
|o.oo..           |
| *..o            |
|..*+. .          |
|+++= . .         |
|oE=.+ o S        |
|.o B . + o       |
|+ B o o o        |
|*+o= =           |
|=o=*= .          |
+----[SHA256]-----+
hduser@Maneesh:/$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
hduser@Maneesh:/$
```

2. Now you need to disable IPv6. Open the /etc/sysctl.conf file and add the following lines to the end of the file and save it. (One way of opening the file is sudo nano /etc/sysctl.conf, after you add the lines you need to press Ctrl+X, Shift Y and Enter)



```
hduser@Maneesh:/$ sudo nano /etc/sysctl.conf
[sudo] password for hduser:
```

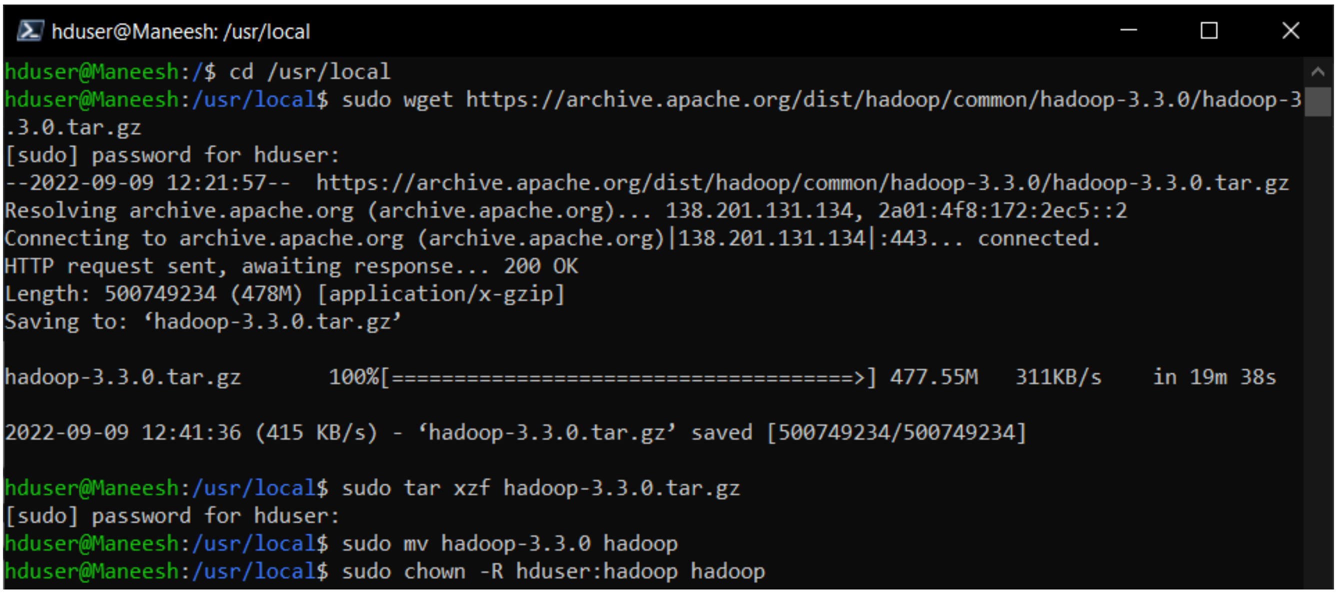
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1



```
  GNU nano 6.2                                 /etc/sysctl.conf *
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

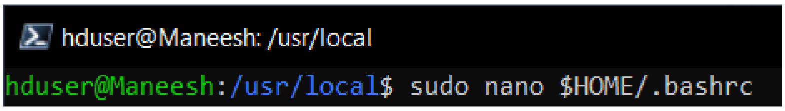
We need to restart the system at this point.

3. Now we download hadoop
   - cd /usr/local
   - sudo wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
   - sudo tar xzf hadoop-3.3.0.tar.gz
   - sudo mv hadoop-3.3.0 hadoop
   - sudo chown -R hduser:hadoop hadoop



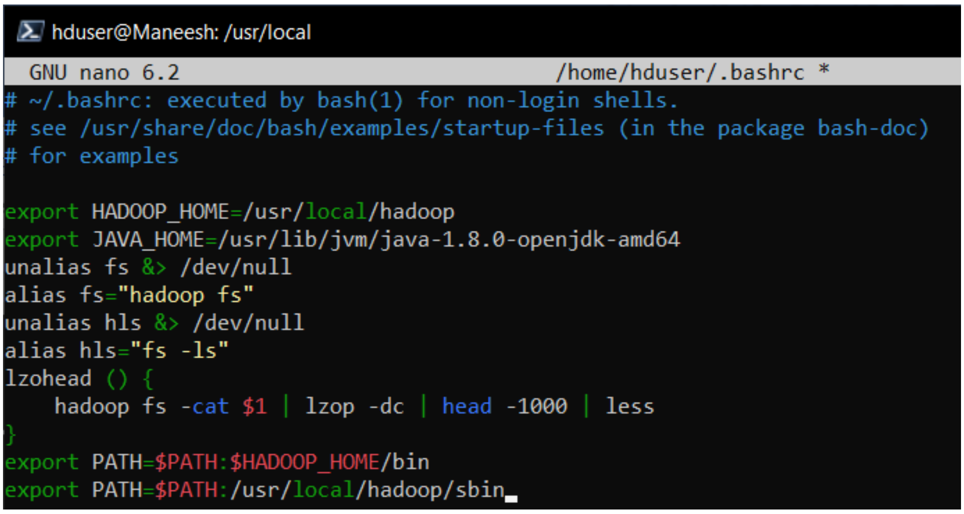
```
hduser@Maneesh:/$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3
.3.0.tar.gz
[sudo] password for hduser:
--2022-09-09 12:21:57--  https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 500749234 (478M) [application/x-gzip]
Saving to: ‘hadoop-3.3.0.tar.gz’

hadoop-3.3.0.tar.gz      100%[====================================>] 477.55M   311KB/s    in 19m 38s

2022-09-09 12:41:36 (415 KB/s) - ‘hadoop-3.3.0.tar.gz’ saved [500749234/500749234]

hduser@Maneesh:/usr/local$ sudo tar xzf hadoop-3.3.0.tar.gz
[sudo] password for hduser:
hduser@Maneesh:/usr/local$ sudo mv hadoop-3.3.0 hadoop
hduser@Maneesh:/usr/local$ sudo chown -R hduser:hadoop hadoop
```

4. Now open sudo nano $HOME/.bashrc and add the following line



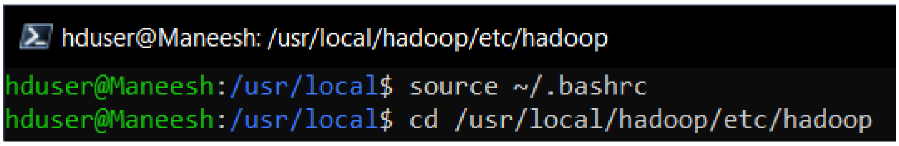
```
hduser@Maneesh:/usr/local$ sudo nano $HOME/.bashrc
```

```
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"
lzohead () {
   hadoop fs -cat $1 | lzop -dc | head -1000 | less
}
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:/usr/local/hadoop/sbin
```



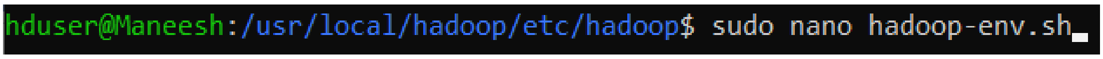
```
hduser@Maneesh: /usr/local
  GNU nano 6.2                          /home/hduser/.bashrc *
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"
lzohead () {
    hadoop fs -cat $1 | lzop -dc | head -1000 | less
}
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:/usr/local/hadoop/sbin
```

5. Enter following commands:
   - ➢ source ~/.bashrc
   - ➢ cd /usr/local/hadoop/etc/hadoop



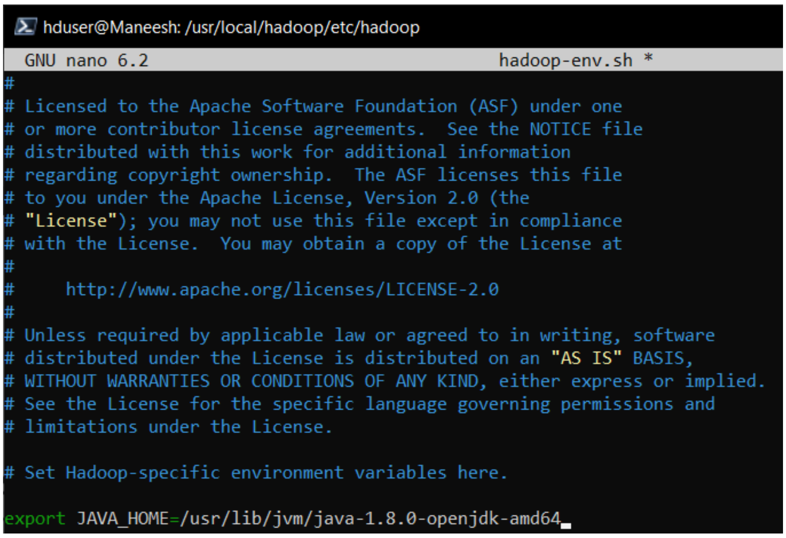
```
hduser@Maneesh: /usr/local/hadoop/etc/hadoop
hduser@Maneesh:/usr/local$ source ~/.bashrc
hduser@Maneesh:/usr/local$ cd /usr/local/hadoop/etc/hadoop
```

6. Add the following line to sudo nano hadoop-env.sh



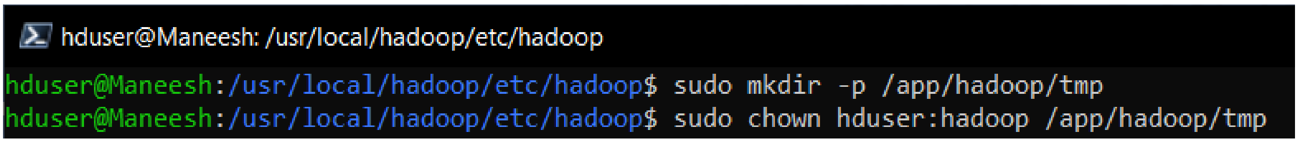
```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo nano hadoop-env.sh
```

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64



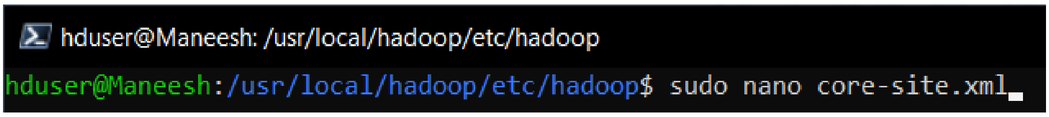
```
hduser@Maneesh: /usr/local/hadoop/etc/hadoop
  GNU nano 6.2                          hadoop-env.sh *
#
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you 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.

# Set Hadoop-specific environment variables here.

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

7. Run the following commands:
   - sudo mkdir -p /app/hadoop/tmp
   - sudo chown hduser:hadoop /app/hadoop/tmp



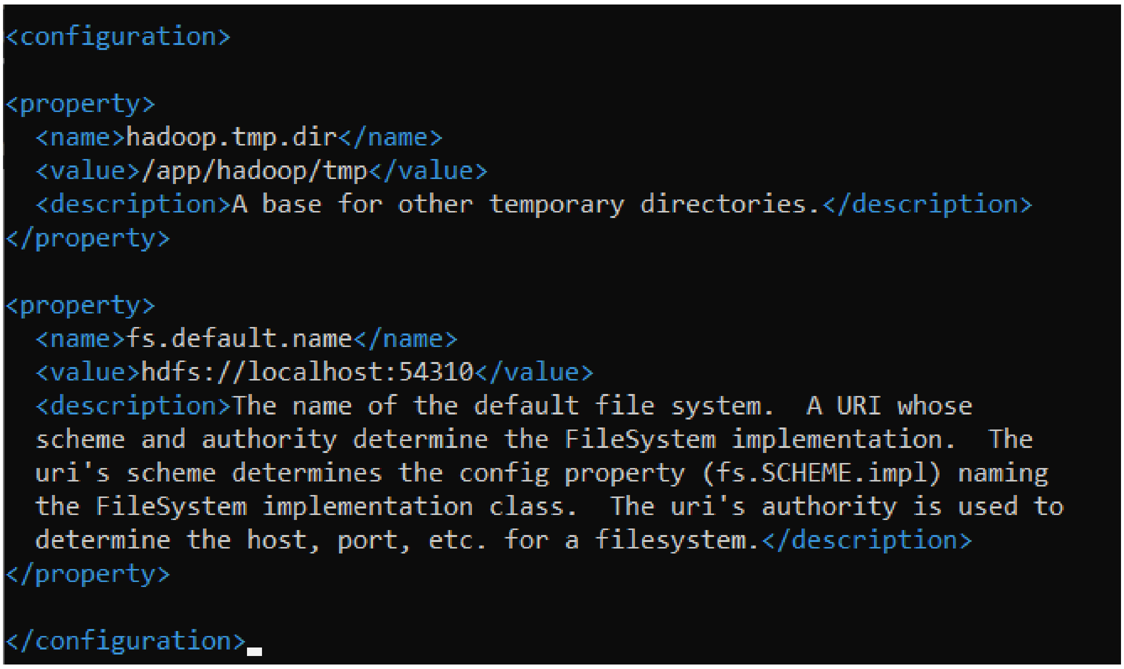
```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p /app/hadoop/tmp
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo chown hduser:hadoop /app/hadoop/tmp
```

8. Make the following changes in sudo nano core-site.xml (this file is present in /usr/local/hadoop/etc/hadoop) Add the following between <configuration> and </configuration>



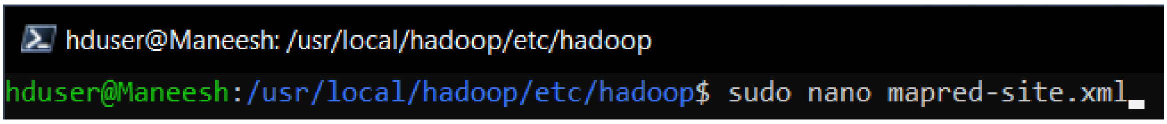
```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo nano core-site.xml
```

```
<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:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>
```



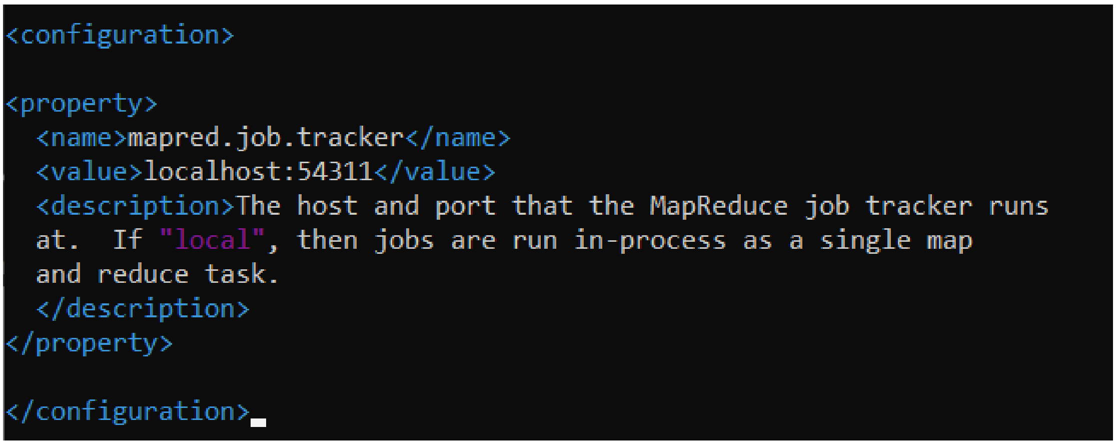
```
<configuration>

<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:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>

</configuration>
```

9. In the file sudo nano mapred-site.xml Add the following between <configuration> and </configuration>



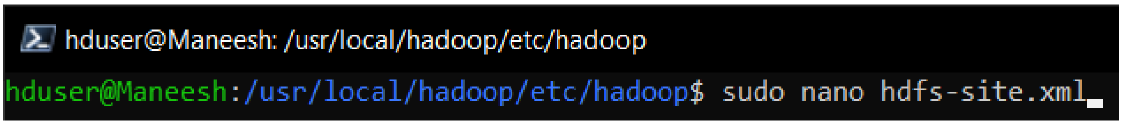
```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo nano mapred-site.xml
```

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>
```



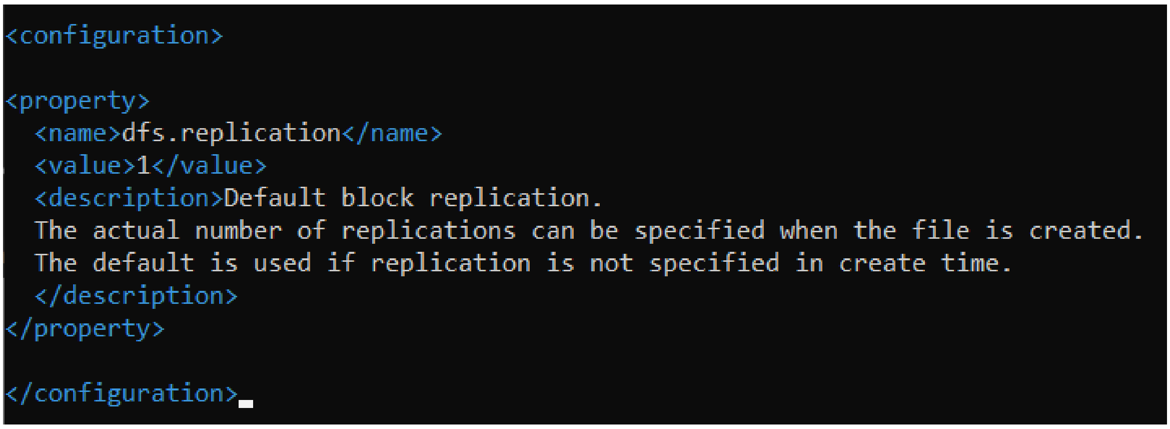
```
<configuration>

<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>

</configuration>
```

10. In the file sudo nano hdfs-site.xml Add the following between <configuration> and </configuration>



```
hduser@Maneesh: /usr/local/hadoop/etc/hadoop
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo nano hdfs-site.xml
```

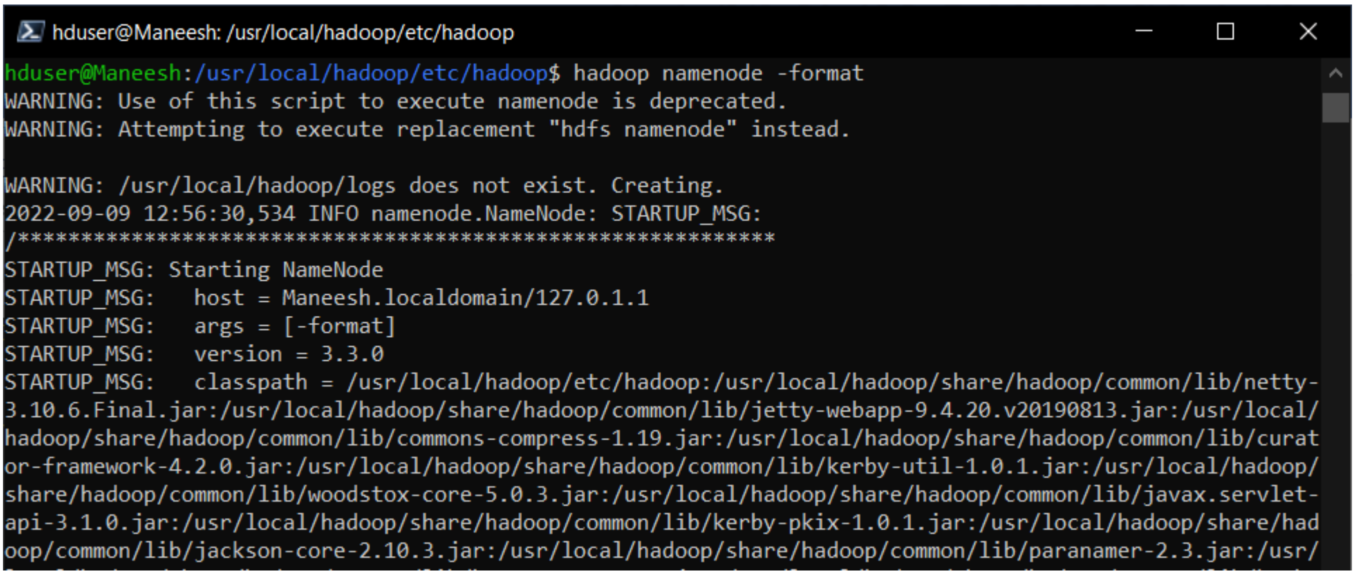
```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>
```



```
<configuration>

<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>

</configuration>
```

11. Finally we format namenode by the following commands:

- hadoop namenode -format



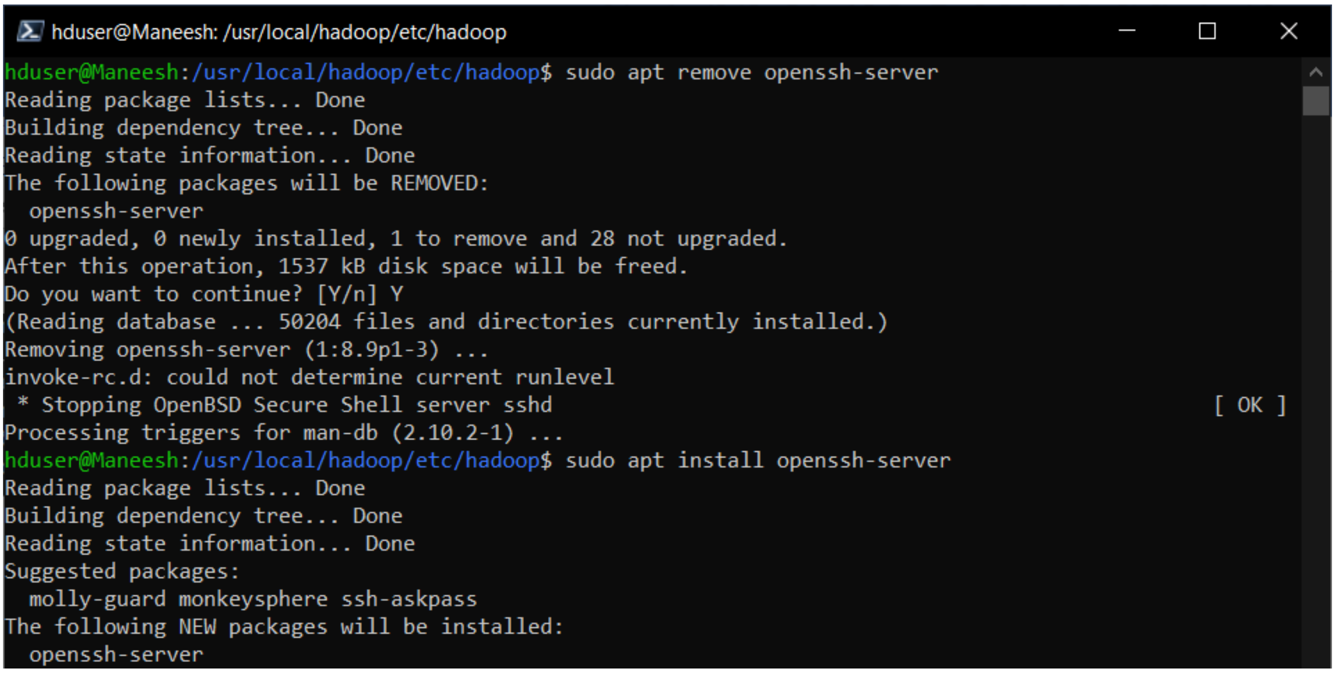
hduser@Maneesh: /usr/local/hadoop/etc/hadoop

```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ hadoop namenode -format
WARNING: Use of this script to execute namenode is deprecated.
WARNING: Attempting to execute replacement "hdfs namenode" instead.

WARNING: /usr/local/hadoop/logs does not exist. Creating.
2022-09-09 12:56:30,534 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Maneesh.localdomain/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/netty-
3.10.6.Final.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-webapp-9.4.20.v20190813.jar:/usr/local/
hadoop/share/hadoop/common/lib/commons-compress-1.19.jar:/usr/local/hadoop/share/hadoop/common/lib/curat
or-framework-4.2.0.jar:/usr/local/hadoop/share/hadoop/common/lib/kerby-util-1.0.1.jar:/usr/local/hadoop/
share/hadoop/common/lib/woodstox-core-5.0.3.jar:/usr/local/hadoop/share/hadoop/common/lib/javax.servlet-
api-3.1.0.jar:/usr/local/hadoop/share/hadoop/common/lib/kerby-pkix-1.0.1.jar:/usr/local/hadoop/share/had
oop/common/lib/jackson-core-2.10.3.jar:/usr/local/hadoop/share/hadoop/common/lib/paranamer-2.3.jar:/usr/
```

THUS WE SUCCESSFULLY INSTALLED HADOOP.

Now to start hadoop, we need to run command:

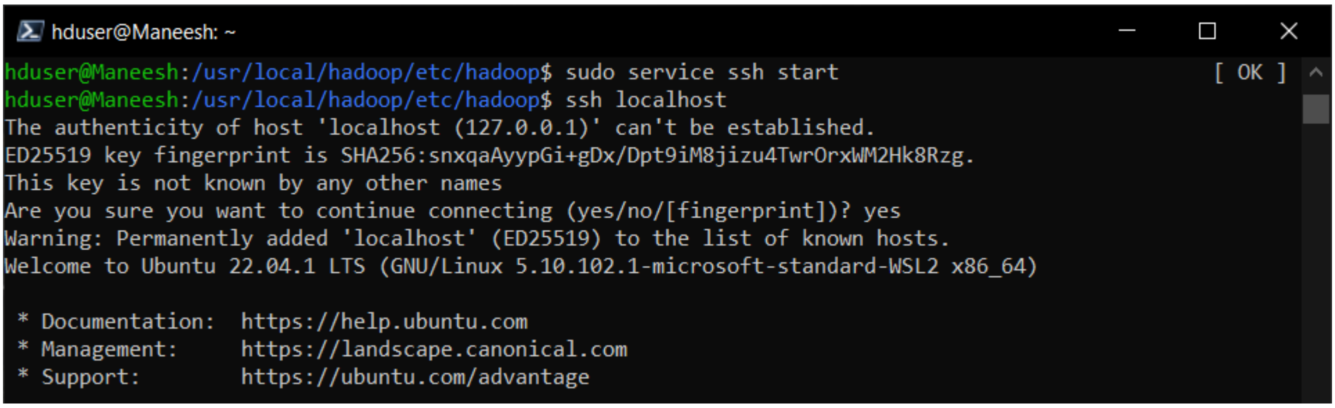
- sudo apt remove openssh-server
- sudo apt install openssh-server



hduser@Maneesh: /usr/local/hadoop/etc/hadoop

```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo apt remove openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  openssh-server
0 upgraded, 0 newly installed, 1 to remove and 28 not upgraded.
After this operation, 1537 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 50204 files and directories currently installed.)
Removing openssh-server (1:8.9p1-3) ...
invoke-rc.d: could not determine current runlevel
 * Stopping OpenBSD Secure Shell server sshd                                                    [ OK ]
Processing triggers for man-db (2.10.2-1) ...
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  openssh-server
```

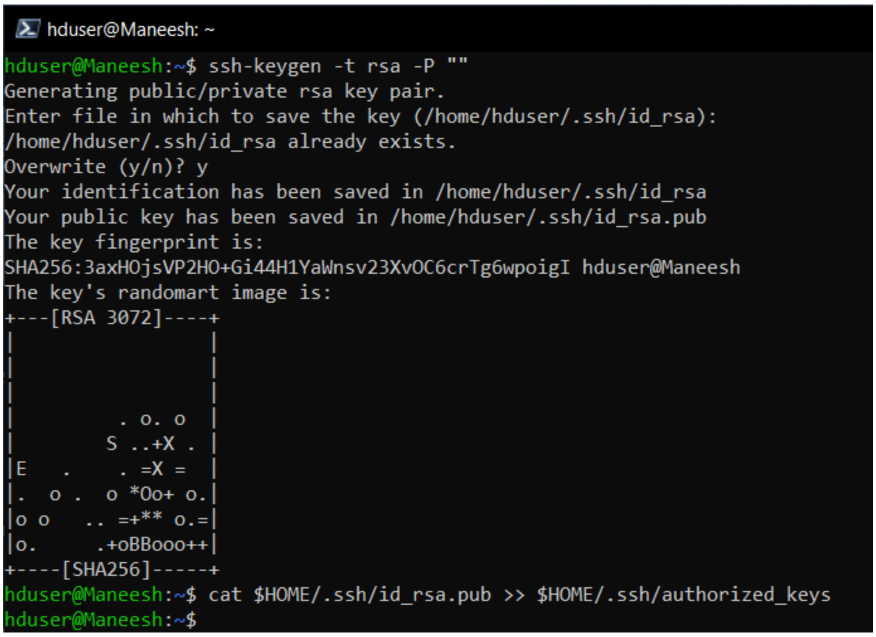
- sudo service ssh start
- ssh localhost



hduser@Maneesh: ~

```
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ sudo service ssh start                             [ OK ]
hduser@Maneesh:/usr/local/hadoop/etc/hadoop$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:snxqaAyypGi+gDx/Dpt9iM8jizu4TwrOrxWM2Hk8Rzg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

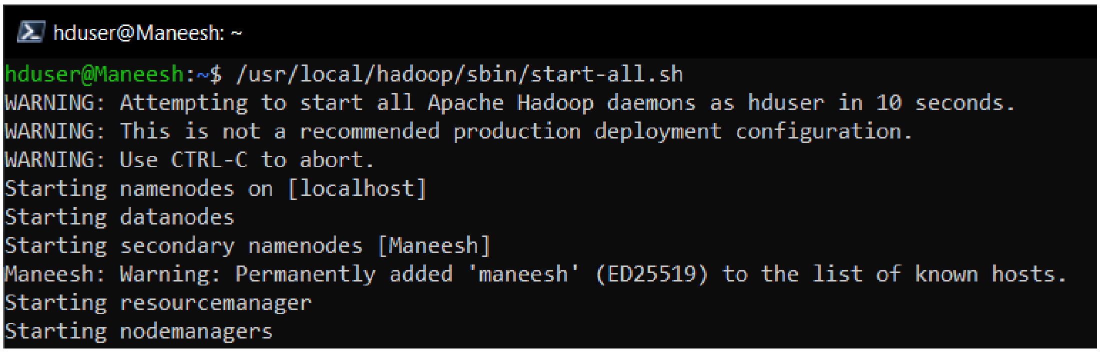
- ssh-keygen -t rsa -P ""
- cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys



```
hduser@Maneesh: ~
hduser@Maneesh:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
/home/hduser/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/hduser/.ssh/id_rsa
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3axHOjsVP2HO+Gi44H1YaWnsv23XvOC6crTg6wpoigI hduser@Maneesh
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|                 |
|         . o. o  |
|        S ..+X . |
|E   .    . =X =  |
|.  o .  o *Oo+ o.|
|o o   .. =+** o.=|
|o.     .+oBBooo++|
+----[SHA256]-----+
hduser@Maneesh:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
hduser@Maneesh:~$
```

This command is to start hadoop services:

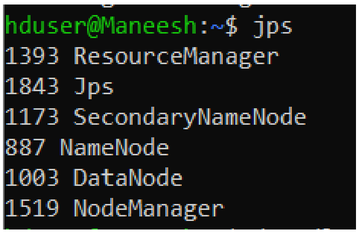
- /usr/local/hadoop/sbin/start-all.sh



```
hduser@Maneesh: ~
hduser@Maneesh:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Maneesh]
Maneesh: Warning: Permanently added 'maneesh' (ED25519) to the list of known hosts.
Starting resourcemanager
Starting nodemanagers
```

This command is to check that all hadoop services are running (6 services should appear):

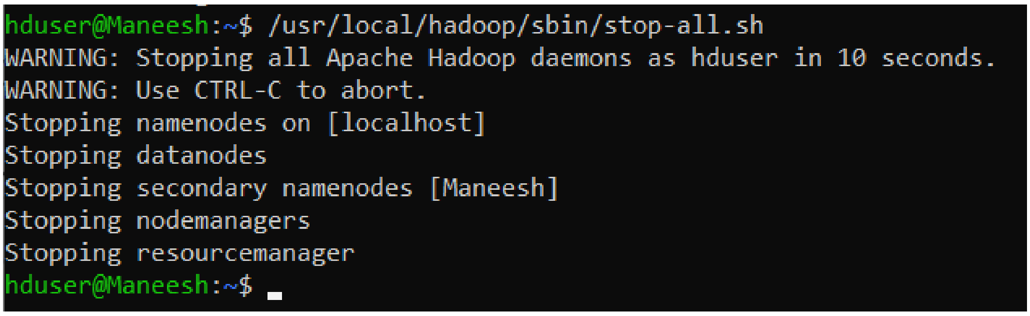
- jps



```
hduser@Maneesh:~$ jps
1393 ResourceManager
1843 Jps
1173 SecondaryNameNode
887 NameNode
1003 DataNode
1519 NodeManager
```

This command is to stop hadoop services:

- /usr/local/hadoop/sbin/stop-all.sh



```
hduser@Maneesh:~$ /usr/local/hadoop/sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [Maneesh]
Stopping nodemanagers
Stopping resourcemanager
hduser@Maneesh:~$ _
```

Practical No: 2

Aim: Download and install Spark.

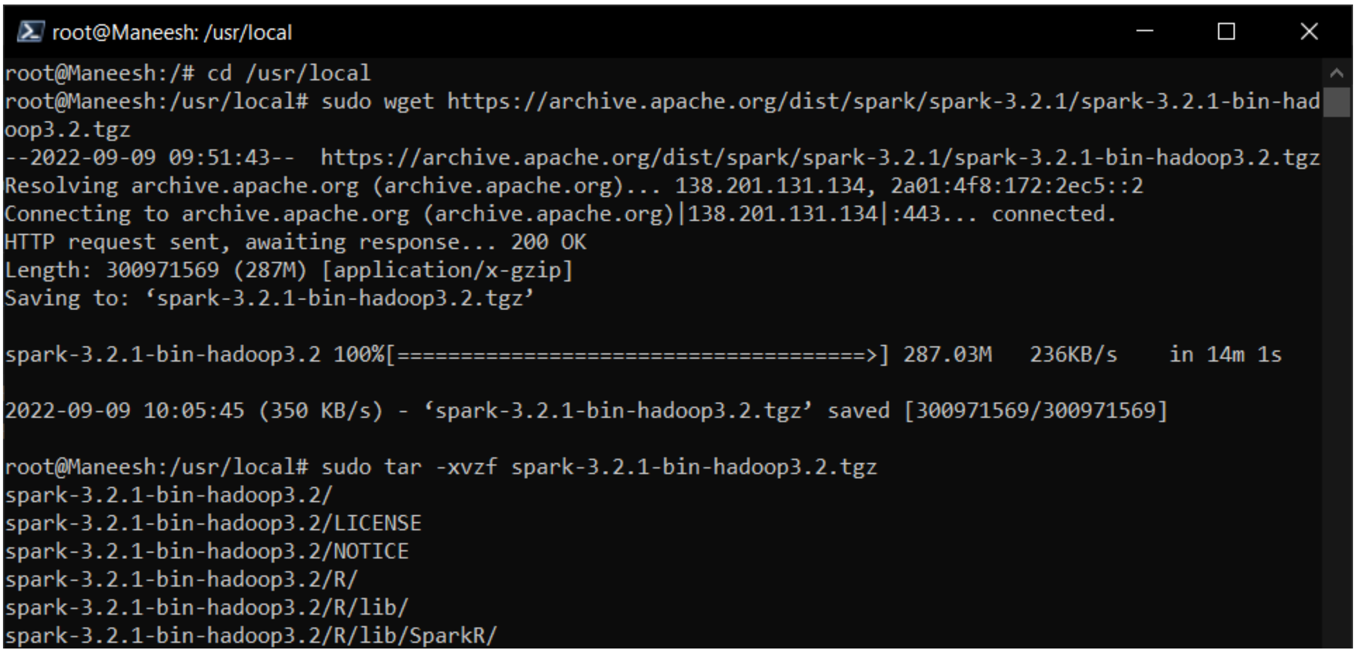
Theory:

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the worlds of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

From its humble beginnings in the AMPLab at U.C. Berkeley in 2009, Apache Spark has become one of the key big data distributed processing frameworks in the world. Spark can be deployed in a variety of ways, provides native bindings for the Java, Scala, Python, and R programming languages, and supports SQL, streaming data, machine learning, and graph processing. You’ll find it used by banks, telecommunications companies, games companies, governments, and all of the major tech giants such as Apple, Facebook, IBM, and Microsoft.

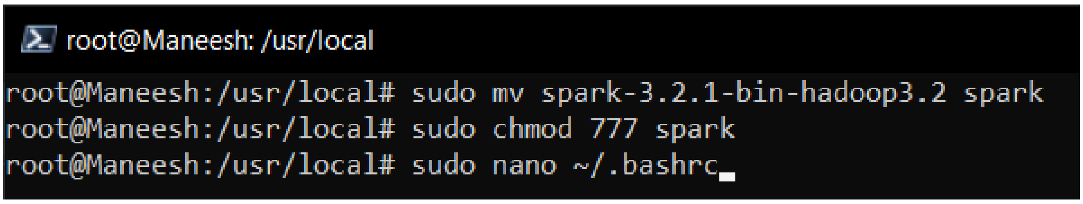
Steps:

1. Download and unzip spark.
   - ➢ cd /usr/local
   - ➢ sudo wget https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
   - ➢ sudo tar -xvzf spark-3.2.1-bin-hadoop3.2.tgz



```
root@Maneesh:/# cd /usr/local
root@Maneesh:/usr/local# sudo wget https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-had
oop3.2.tgz
--2022-09-09 09:51:43--  https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 300971569 (287M) [application/x-gzip]
Saving to: ‘spark-3.2.1-bin-hadoop3.2.tgz’

spark-3.2.1-bin-hadoop3.2 100%[====================================>] 287.03M   236KB/s    in 14m 1s

2022-09-09 10:05:45 (350 KB/s) - ‘spark-3.2.1-bin-hadoop3.2.tgz’ saved [300971569/300971569]

root@Maneesh:/usr/local# sudo tar -xvzf spark-3.2.1-bin-hadoop3.2.tgz
spark-3.2.1-bin-hadoop3.2/
spark-3.2.1-bin-hadoop3.2/LICENSE
spark-3.2.1-bin-hadoop3.2/NOTICE
spark-3.2.1-bin-hadoop3.2/R/
spark-3.2.1-bin-hadoop3.2/R/lib/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/
```

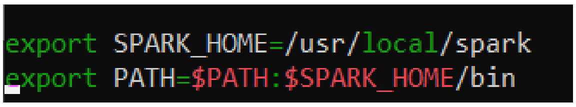
2. move the unzip spark folder to “spark” folder, give permission to folder and open ~/.bashrc.
   - ➢ sudo mv spark-3.2.1-bin-hadoop3.2 spark
   - ➢ sudo chmod 777 spark
   - ➢ sudo nano ~/.bashrc



```
root@Maneesh:/usr/local# sudo mv spark-3.2.1-bin-hadoop3.2 spark
root@Maneesh:/usr/local# sudo chmod 777 spark
root@Maneesh:/usr/local# sudo nano ~/.bashrc
```

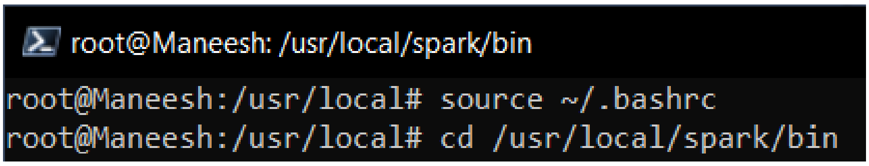
3. add spark path location in bashrc.

export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin



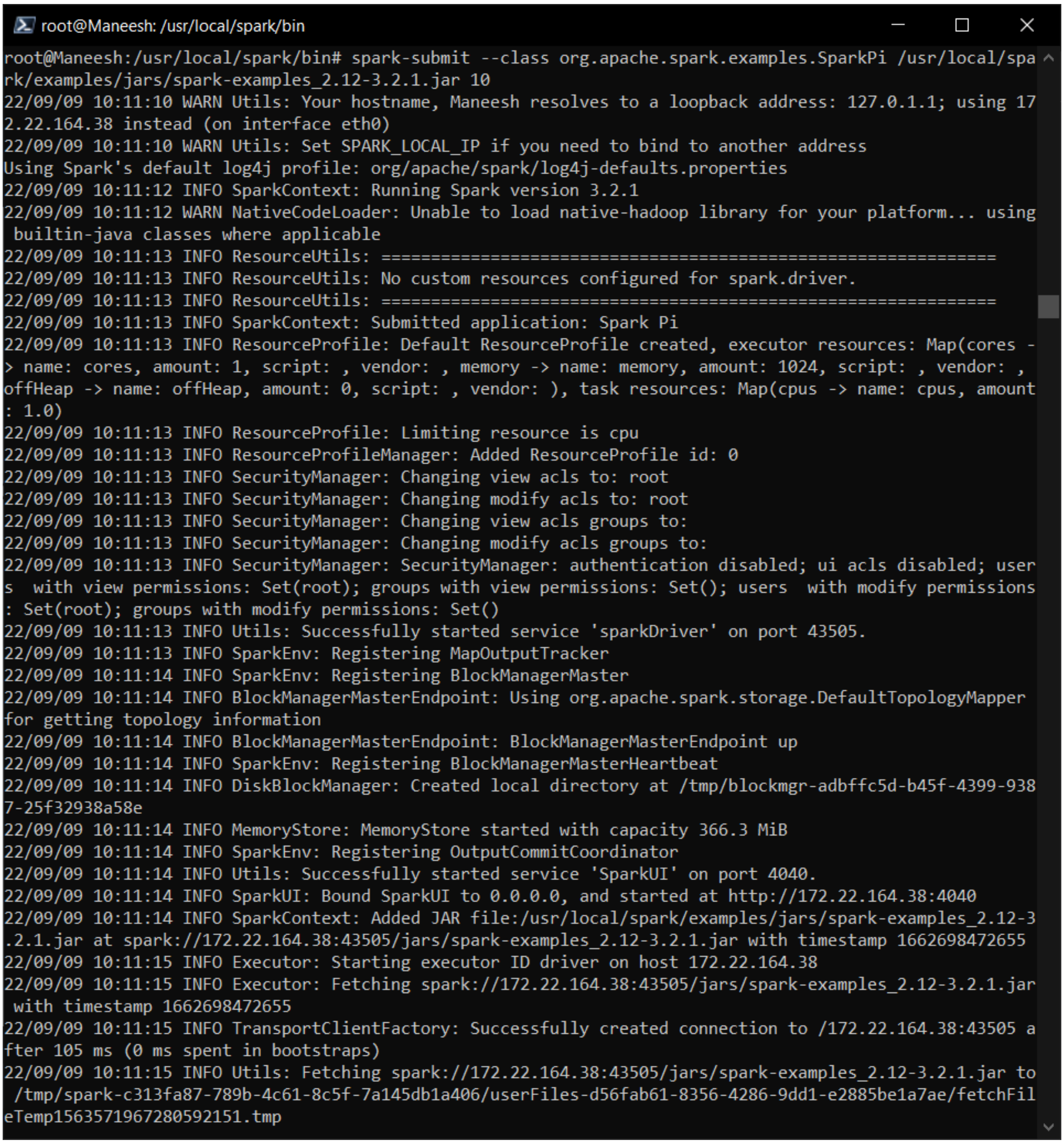
```
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
```

4. save bashrc.
   - ➢ source ~/.bashrc
   - ➢ cd /usr/local/spark/bin



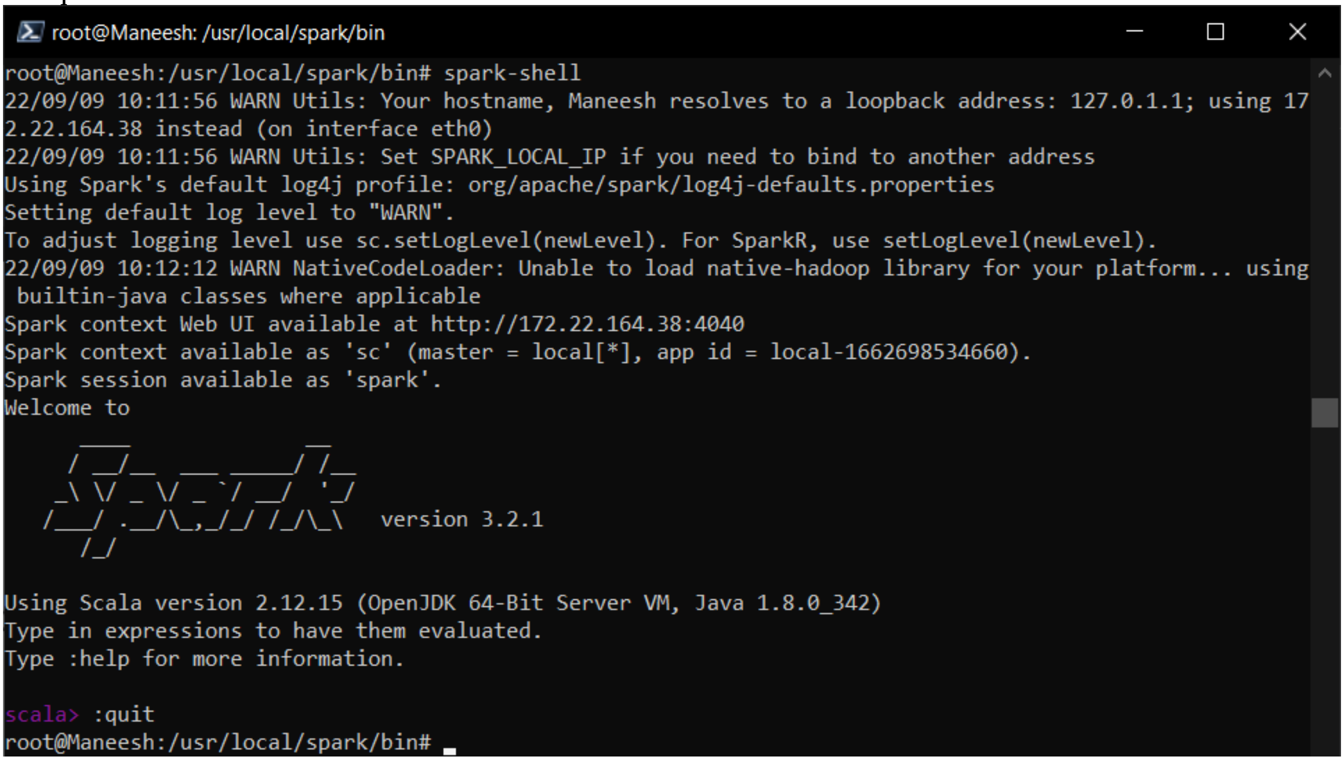
```
root@Maneesh:/usr/local# source ~/.bashrc
root@Maneesh:/usr/local# cd /usr/local/spark/bin
```

5. Run spark.
   - ➢ spark-submit --class org.apache.spark.examples.SparkPi /usr/local/spark/examples/jars/spark-examples_2.12-3.2.1.jar 10



```
root@Maneesh:/usr/local/spark/bin# spark-submit --class org.apache.spark.examples.SparkPi /usr/local/spa
rk/examples/jars/spark-examples_2.12-3.2.1.jar 10
22/09/09 10:11:10 WARN Utils: Your hostname, Maneesh resolves to a loopback address: 127.0.1.1; using 17
2.22.164.38 instead (on interface eth0)
22/09/09 10:11:10 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
22/09/09 10:11:12 INFO SparkContext: Running Spark version 3.2.1
22/09/09 10:11:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
 builtin-java classes where applicable
22/09/09 10:11:13 INFO ResourceUtils: ==============================================================
22/09/09 10:11:13 INFO ResourceUtils: No custom resources configured for spark.driver.
22/09/09 10:11:13 INFO ResourceUtils: ==============================================================
22/09/09 10:11:13 INFO SparkContext: Submitted application: Spark Pi
22/09/09 10:11:13 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -
> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: ,
offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount
: 1.0)
22/09/09 10:11:13 INFO ResourceProfile: Limiting resource is cpu
22/09/09 10:11:13 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/09/09 10:11:13 INFO SecurityManager: Changing view acls to: root
22/09/09 10:11:13 INFO SecurityManager: Changing modify acls to: root
22/09/09 10:11:13 INFO SecurityManager: Changing view acls groups to:
22/09/09 10:11:13 INFO SecurityManager: Changing modify acls groups to:
22/09/09 10:11:13 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; user
s  with view permissions: Set(root); groups with view permissions: Set(); users  with modify permissions
: Set(root); groups with modify permissions: Set()
22/09/09 10:11:13 INFO Utils: Successfully started service 'sparkDriver' on port 43505.
22/09/09 10:11:13 INFO SparkEnv: Registering MapOutputTracker
22/09/09 10:11:14 INFO SparkEnv: Registering BlockManagerMaster
22/09/09 10:11:14 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper
for getting topology information
22/09/09 10:11:14 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
22/09/09 10:11:14 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
22/09/09 10:11:14 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-adbffc5d-b45f-4399-938
7-25f32938a58e
22/09/09 10:11:14 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
22/09/09 10:11:14 INFO SparkEnv: Registering OutputCommitCoordinator
22/09/09 10:11:14 INFO Utils: Successfully started service 'SparkUI' on port 4040.
22/09/09 10:11:14 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://172.22.164.38:4040
22/09/09 10:11:14 INFO SparkContext: Added JAR file:/usr/local/spark/examples/jars/spark-examples_2.12-3
.2.1.jar at spark://172.22.164.38:43505/jars/spark-examples_2.12-3.2.1.jar with timestamp 1662698472655
22/09/09 10:11:15 INFO Executor: Starting executor ID driver on host 172.22.164.38
22/09/09 10:11:15 INFO Executor: Fetching spark://172.22.164.38:43505/jars/spark-examples_2.12-3.2.1.jar
 with timestamp 1662698472655
22/09/09 10:11:15 INFO TransportClientFactory: Successfully created connection to /172.22.164.38:43505 a
fter 105 ms (0 ms spent in bootstraps)
22/09/09 10:11:15 INFO Utils: Fetching spark://172.22.164.38:43505/jars/spark-examples_2.12-3.2.1.jar to
 /tmp/spark-c313fa87-789b-4c61-8c5f-7a145db1a406/userFiles-d56fab61-8356-4286-9dd1-e2885be1a7ae/fetchFil
eTemp1563571967280592151.tmp
```

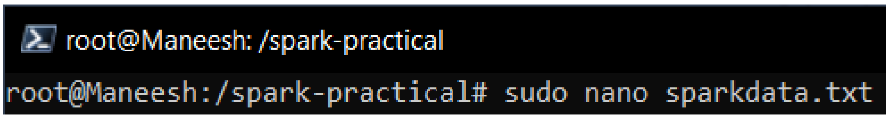
- ➢ spark-shell
  :quit



```
root@Maneesh:/usr/local/spark/bin# spark-shell
22/09/09 10:11:56 WARN Utils: Your hostname, Maneesh resolves to a loopback address: 127.0.1.1; using 17
2.22.164.38 instead (on interface eth0)
22/09/09 10:11:56 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/09/09 10:12:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
 builtin-java classes where applicable
Spark context Web UI available at http://172.22.164.38:4040
Spark context available as 'sc' (master = local[*], app id = local-1662698534660).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_342)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :quit
root@Maneesh:/usr/local/spark/bin#
```

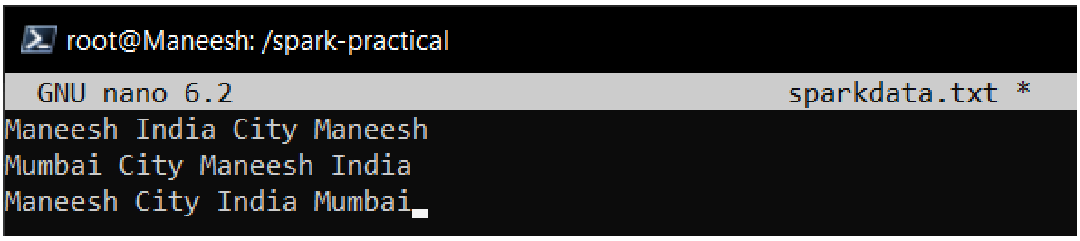
Word count problem using SPARK Scala:

- ➢ sudo nano sparkdata.txt



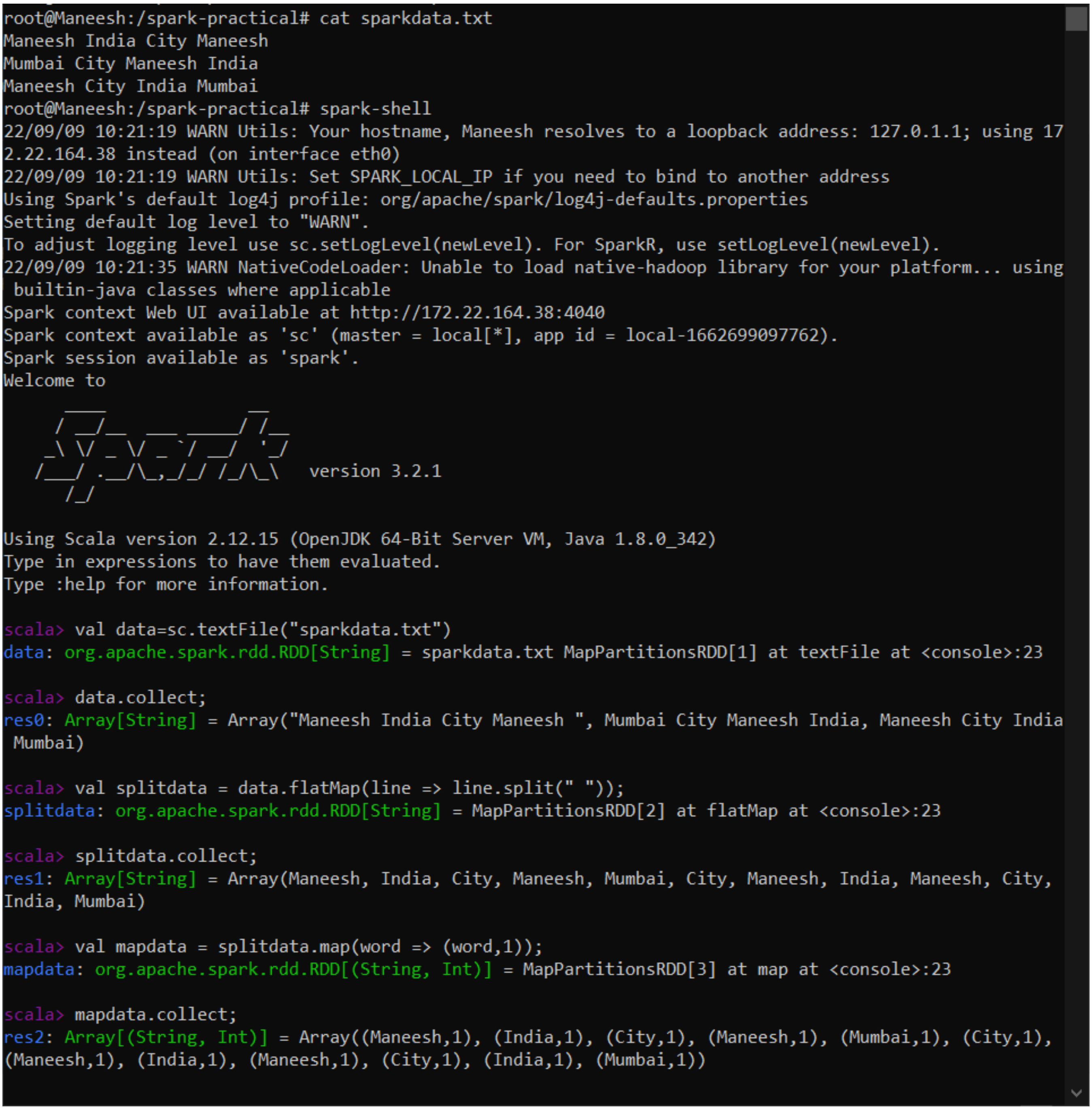
```
root@Maneesh:/spark-practical# sudo nano sparkdata.txt
```

Maneesh India City Maneesh
Mumbai City Maneesh India
Maneesh City India Mumbai

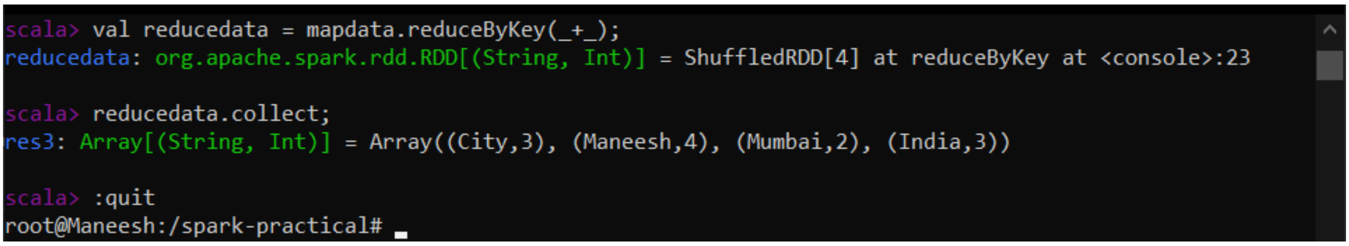


```
  GNU nano 6.2                                  sparkdata.txt *
Maneesh India City Maneesh
Mumbai City Maneesh India
Maneesh City India Mumbai
```

1. cat sparkdata.txt
2. spark-shell
3. val data=sc.textFile("sparkdata.txt")
4. data.collect;
5. val splitdata = data.flatMap(line => line.split(" "));
6. splitdata.collect;
7. val mapdata = splitdata.map(word => (word,1));
8. mapdata.collect;
9. val reducedata = mapdata.reduceByKey(_+_);
10. reducedata.collect;
11. :quit



```
root@Maneesh:/spark-practical# cat sparkdata.txt
Maneesh India City Maneesh
Mumbai City Maneesh India
Maneesh City India Mumbai
root@Maneesh:/spark-practical# spark-shell
22/09/09 10:21:19 WARN Utils: Your hostname, Maneesh resolves to a loopback address: 127.0.1.1; using 17
2.22.164.38 instead (on interface eth0)
22/09/09 10:21:19 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/09/09 10:21:35 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
 builtin-java classes where applicable
Spark context Web UI available at http://172.22.164.38:4040
Spark context available as 'sc' (master = local[*], app id = local-1662699097762).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_342)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val data=sc.textFile("sparkdata.txt")
data: org.apache.spark.rdd.RDD[String] = sparkdata.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> data.collect;
res0: Array[String] = Array("Maneesh India City Maneesh ", Mumbai City Maneesh India, Maneesh City India
 Mumbai)

scala> val splitdata = data.flatMap(line => line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:23

scala> splitdata.collect;
res1: Array[String] = Array(Maneesh, India, City, Maneesh, Mumbai, City, Maneesh, India, Maneesh, City,
India, Mumbai)

scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:23

scala> mapdata.collect;
res2: Array[(String, Int)] = Array((Maneesh,1), (India,1), (City,1), (Maneesh,1), (Mumbai,1), (City,1),
(Maneesh,1), (India,1), (Maneesh,1), (City,1), (India,1), (Mumbai,1))
```



```
scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> reducedata.collect;
res3: Array[(String, Int)] = Array((City,3), (Maneesh,4), (Mumbai,2), (India,3))

scala> :quit
root@Maneesh:/spark-practical# _
```

Practical No: 3

Aim: Implementing Map-Reduce Program for Word Count problem.

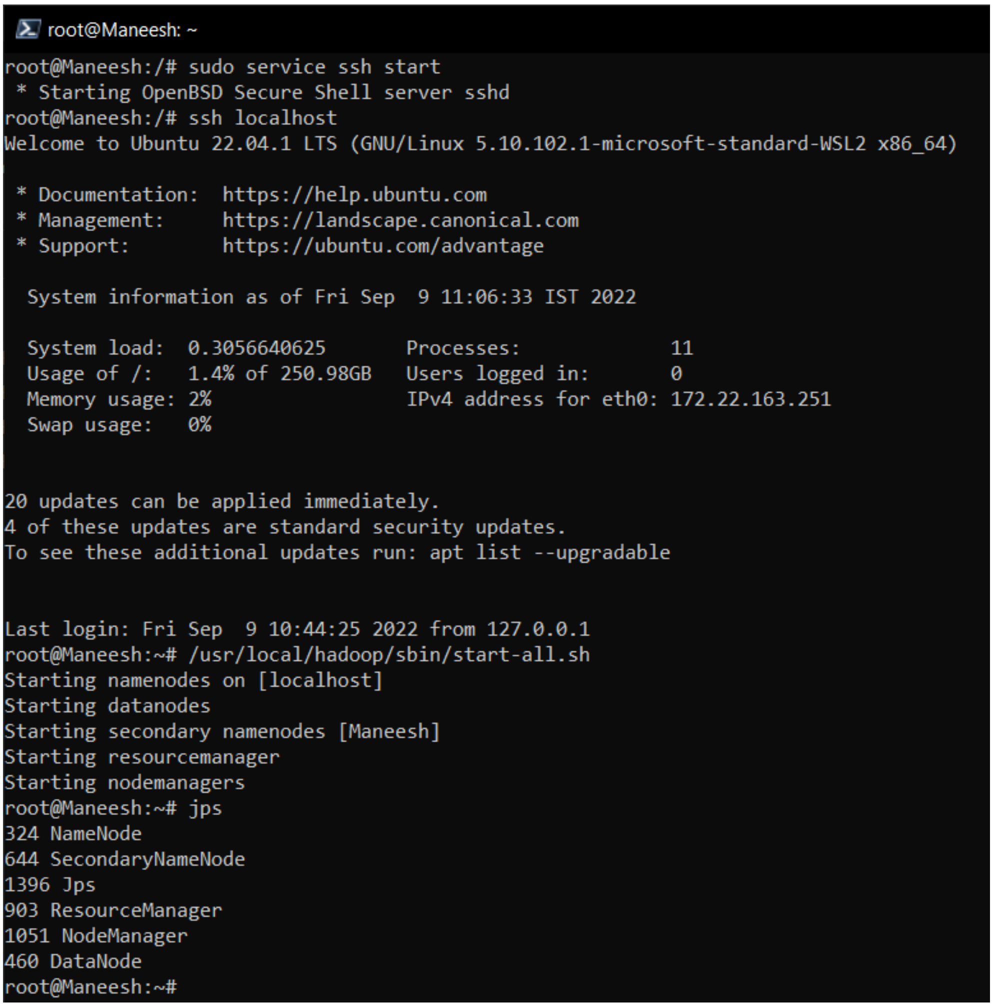
Theory:

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

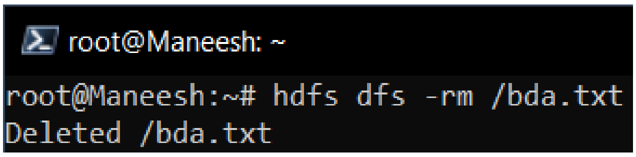
Steps:

1. Start Hadoop
   - ➢ sudo service ssh start
   - ➢ ssh localhost
   - ➢ /usr/local/hadoop/sbin/start-all.sh
   - ➢ jps



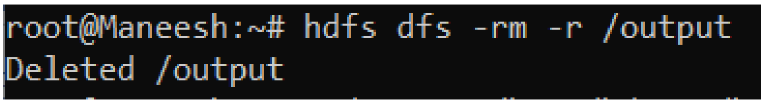
```
root@Maneesh:/# sudo service ssh start
 * Starting OpenBSD Secure Shell server sshd
root@Maneesh:/# ssh localhost
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Sep  9 11:06:33 IST 2022

  System load:  0.3056640625       Processes:             11
  Usage of /:   1.4% of 250.98GB   Users logged in:       0
  Memory usage: 2%                 IPv4 address for eth0: 172.22.163.251
  Swap usage:   0%


20 updates can be applied immediately.
4 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Fri Sep  9 10:44:25 2022 from 127.0.0.1
root@Maneesh:~# /usr/local/hadoop/sbin/start-all.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Maneesh]
Starting resourcemanager
Starting nodemanagers
root@Maneesh:~# jps
324 NameNode
644 SecondaryNameNode
1396 Jps
903 ResourceManager
1051 NodeManager
460 DataNode
root@Maneesh:~#
```

2. To remove existing file from HDFS
   - ➢ hdfs dfs -rm /bda.txt



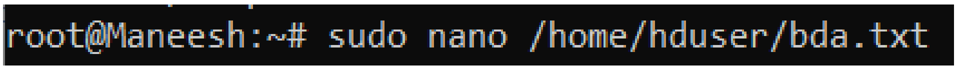
```
root@Maneesh:~# hdfs dfs -rm /bda.txt
Deleted /bda.txt
```

3. Clear Output of previous run at default HDFS location
   - ➢ hdfs dfs -rm -r /output

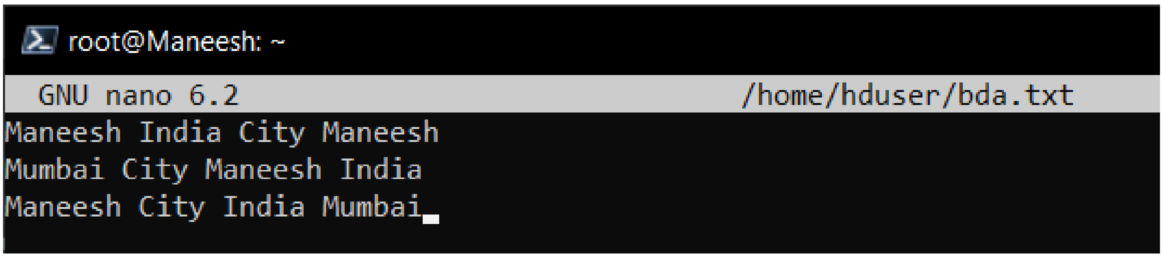


```
root@Maneesh:~# hdfs dfs -rm -r /output
Deleted /output
```

4. Create a text file with some words at local file system (try to include same and repeated words) (Press Ctrl+S and then Ctrl+X)
   - ➢ sudo nano /home/hduser/bda.txt

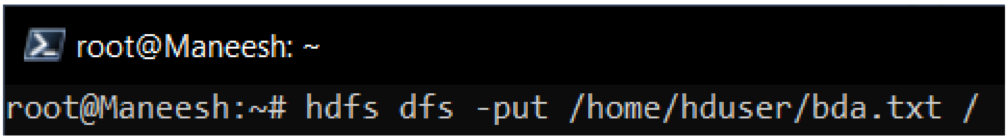


```
root@Maneesh:~# sudo nano /home/hduser/bda.txt
```



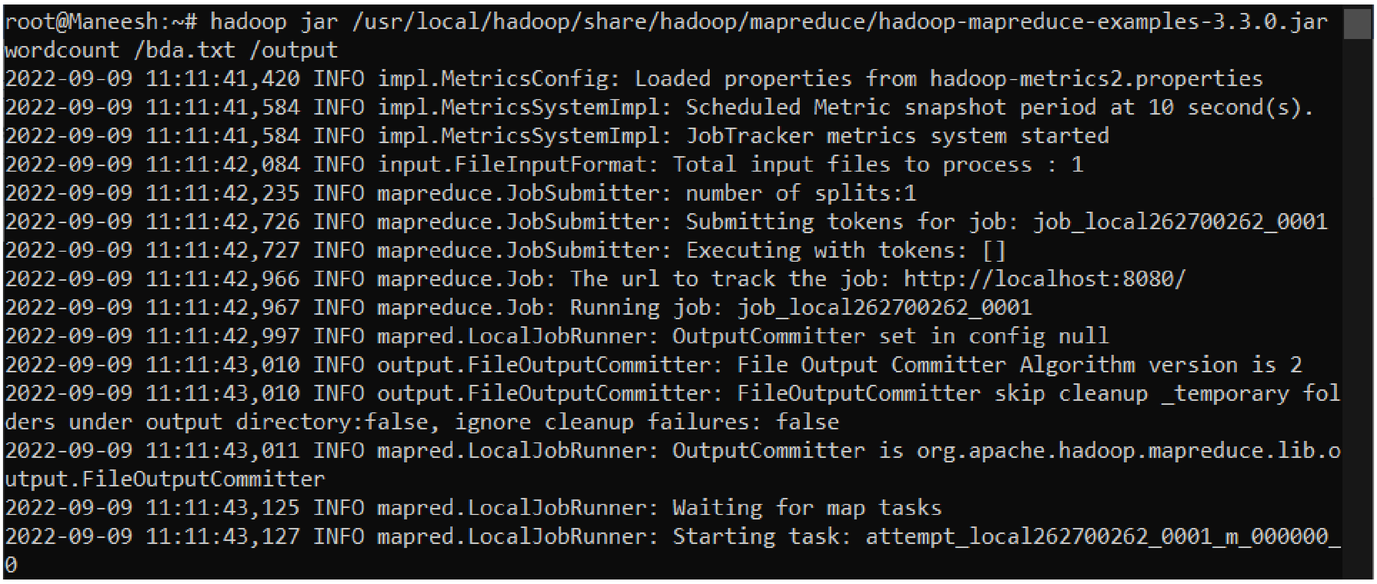
```
  GNU nano 6.2                              /home/hduser/bda.txt
Maneesh India City Maneesh
Mumbai City Maneesh India
Maneesh City India Mumbai
```

5. Move bda.txt file to HDFS
   - ➢ hdfs dfs -put /home/hduser/bda.txt /



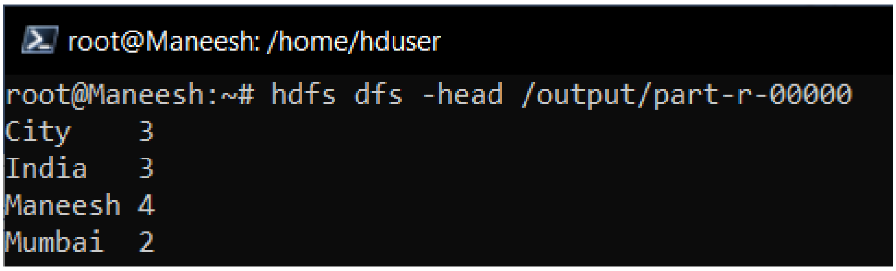
```
root@Maneesh:~# hdfs dfs -put /home/hduser/bda.txt /
```

6. Running MapReduce for wordcount file bda.txt
   - ➢ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /bda.txt /output



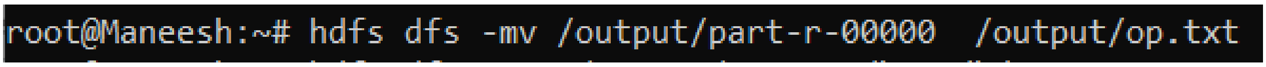
```
root@Maneesh:~# hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar
wordcount /bda.txt /output
2022-09-09 11:11:41,420 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-09-09 11:11:41,584 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-09-09 11:11:41,584 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-09-09 11:11:42,084 INFO input.FileInputFormat: Total input files to process : 1
2022-09-09 11:11:42,235 INFO mapreduce.JobSubmitter: number of splits:1
2022-09-09 11:11:42,726 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local262700262_0001
2022-09-09 11:11:42,727 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-09-09 11:11:42,966 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-09-09 11:11:42,967 INFO mapreduce.Job: Running job: job_local262700262_0001
2022-09-09 11:11:42,997 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-09-09 11:11:43,010 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-09-09 11:11:43,010 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary fol
ders under output directory:false, ignore cleanup failures: false
2022-09-09 11:11:43,011 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.o
utput.FileOutputCommitter
2022-09-09 11:11:43,125 INFO mapred.LocalJobRunner: Waiting for map tasks
2022-09-09 11:11:43,127 INFO mapred.LocalJobRunner: Starting task: attempt_local262700262_0001_m_000000_
0
```

7. Check/display output at default output location
   - ➢ hdfs dfs -head /output/part-r-00000



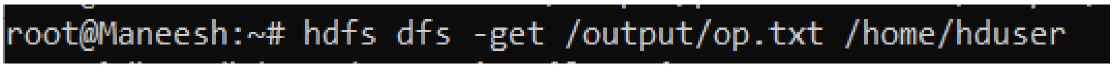
```
root@Maneesh:~# hdfs dfs -head /output/part-r-00000
City    3
India   3
Maneesh 4
Mumbai  2
```

8. To get output in a .txt file in HDFS(optional)
   - ➢ hdfs dfs -mv /output/part-r-00000 /output/op.txt



```
root@Maneesh:~# hdfs dfs -mv /output/part-r-00000  /output/op.txt
```

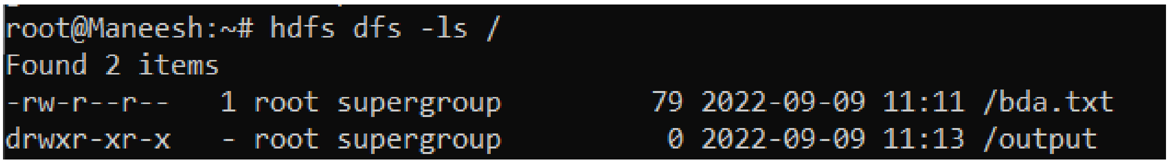
9. To get output in a .txt file in default file location(optional)
   - ➢ hdfs dfs -get /output/op.txt /home/hduser



```
root@Maneesh:~# hdfs dfs -get /output/op.txt /home/hduser
```

To view content of HDFS location, use following command

- ➢ hdfs dfs -ls /



```
root@Maneesh:~# hdfs dfs -ls /
Found 2 items
-rw-r--r--   1 root supergroup         79 2022-09-09 11:11 /bda.txt
drwxr-xr-x   - root supergroup          0 2022-09-09 11:13 /output
```

Practical No: 4

Aim: Install Hive and use Hive Create and store structured databases.

Theory:

Apache Hive is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale. A data warehouse provides a central store of information that can easily be analyzed to make informed, data driven decisions. Hive allows users to read, write, and manage petabytes of data using SQL.

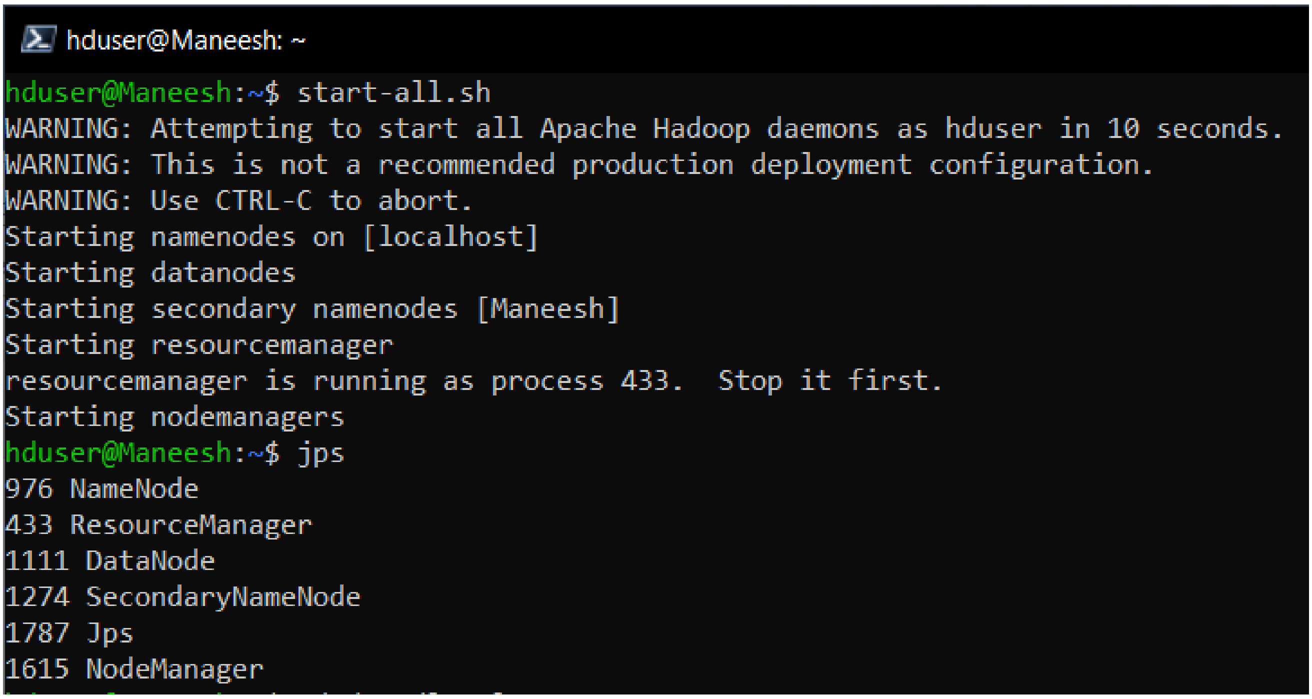
Hive is built on top of Apache Hadoop, which is an open-source framework used to efficiently store and process large datasets. As a result, Hive is closely integrated with Hadoop, and is designed to work quickly on petabytes of data. What makes Hive unique is the ability to query large datasets, leveraging Apache Tez or MapReduce, with a SQL-like interface.

Features of Hive:

- Hive is fast and scalable.
- It provides SQL-like queries (i.e., HQL) that are implicitly transformed to MapReduce or Spark jobs.
- It is capable of analyzing large datasets stored in HDFS.
- It allows different storage types such as plain text, RCFile, and HBase.
- It uses indexing to accelerate queries.
- It can operate on compressed data stored in the Hadoop ecosystem.
- It supports user-defined functions (UDFs) where user can provide its functionality.

Steps:

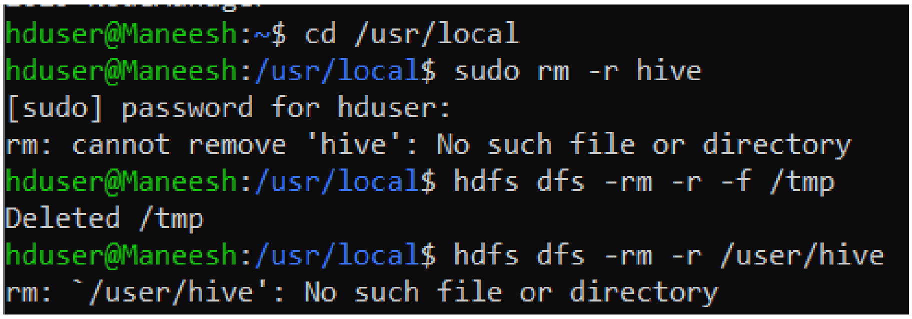
1. First we need to start Hadoop:
   - su hduser
   - sudo service ssh start
   - ssh localhost
   - start-all.sh
   - jps



```
hduser@Maneesh: ~
hduser@Maneesh:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Maneesh]
Starting resourcemanager
resourcemanager is running as process 433.  Stop it first.
Starting nodemanagers
hduser@Maneesh:~$ jps
976 NameNode
433 ResourceManager
1111 DataNode
1274 SecondaryNameNode
1787 Jps
1615 NodeManager
```

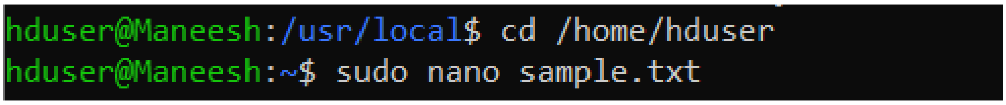
2. Now to uninstall existing version:
   - cd /usr/local
   - sudo rm -r hive

- hdfs dfs -rm -r -f /tmp
- hdfs dfs -rm -r /user/hive



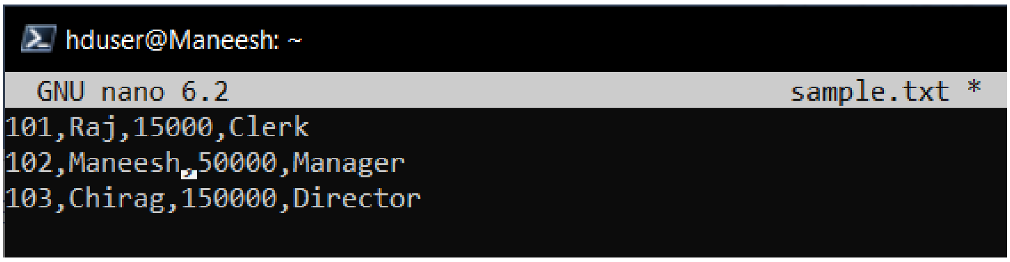
```
hduser@Maneesh:~$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo rm -r hive
[sudo] password for hduser:
rm: cannot remove 'hive': No such file or directory
hduser@Maneesh:/usr/local$ hdfs dfs -rm -r -f /tmp
Deleted /tmp
hduser@Maneesh:/usr/local$ hdfs dfs -rm -r /user/hive
rm: `/user/hive': No such file or directory
```

3. Create a text file sample.txt:
   - cd /home/hduser
   - sudo nano sample.txt



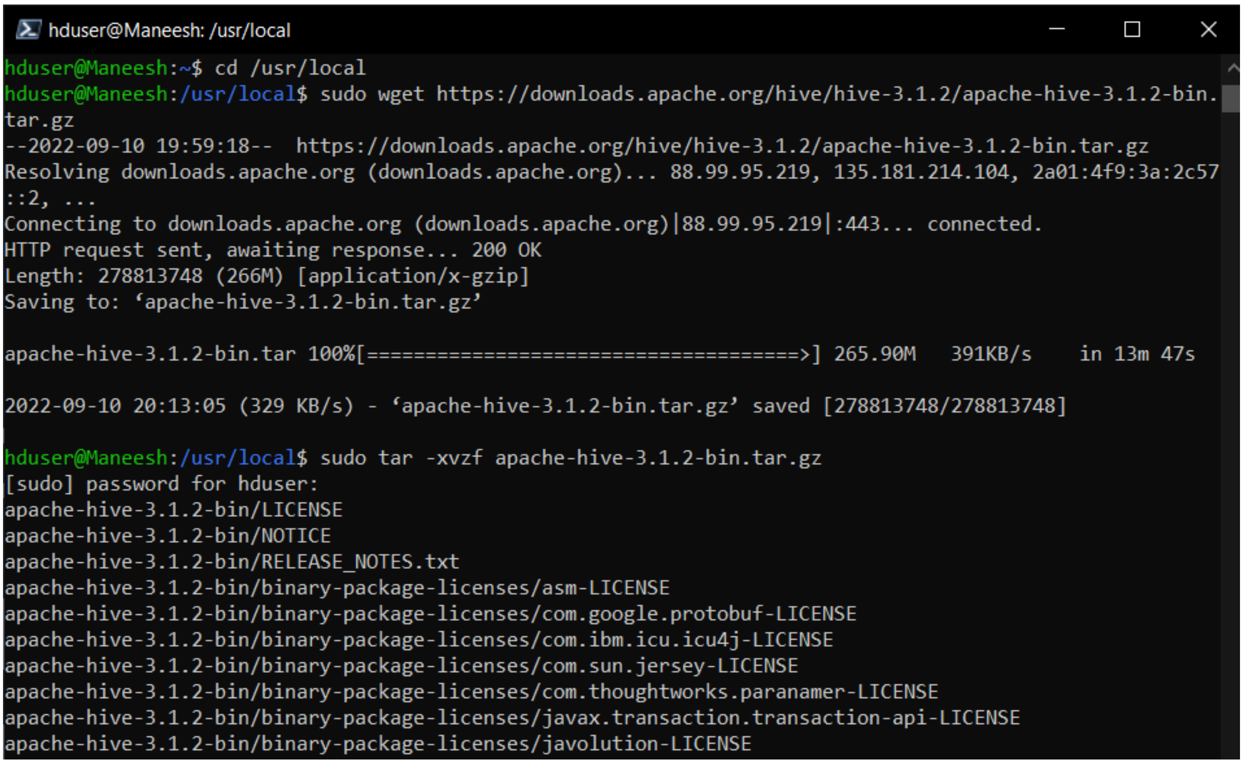
```
hduser@Maneesh:/usr/local$ cd /home/hduser
hduser@Maneesh:~$ sudo nano sample.txt
```

101,Raj,15000,Clerk
102,Maneesh,50000,Manager
103,Chirag,150000,Director



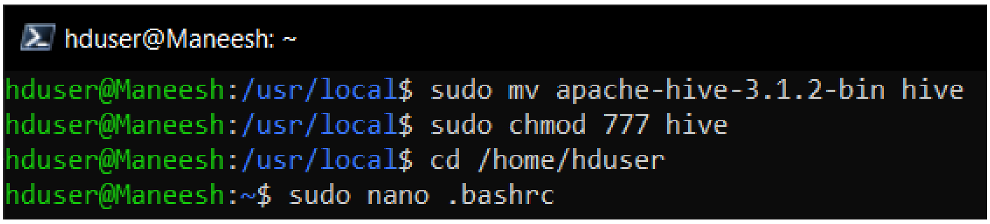
```
hduser@Maneesh: ~
 GNU nano 6.2                                sample.txt *
101,Raj,15000,Clerk
102,Maneesh,50000,Manager
103,Chirag,150000,Director
```

4. Now we download and setup hive:
   - cd /usr/local
   - sudo wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
   - sudo tar -xvzf apache-hive-3.1.2-bin.tar.gz



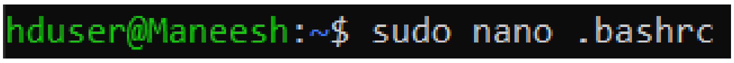
```
hduser@Maneesh: /usr/local
hduser@Maneesh:~$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.
tar.gz
--2022-09-10 19:59:18--  https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.214.104, 2a01:4f9:3a:2c57
::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 278813748 (266M) [application/x-gzip]
Saving to: ‘apache-hive-3.1.2-bin.tar.gz’

apache-hive-3.1.2-bin.tar 100%[====================================>] 265.90M   391KB/s    in 13m 47s

2022-09-10 20:13:05 (329 KB/s) - ‘apache-hive-3.1.2-bin.tar.gz’ saved [278813748/278813748]

hduser@Maneesh:/usr/local$ sudo tar -xvzf apache-hive-3.1.2-bin.tar.gz
[sudo] password for hduser:
apache-hive-3.1.2-bin/LICENSE
apache-hive-3.1.2-bin/NOTICE
apache-hive-3.1.2-bin/RELEASE_NOTES.txt
apache-hive-3.1.2-bin/binary-package-licenses/asm-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.google.protobuf-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.ibm.icu.icu4j-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.sun.jersey-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/com.thoughtworks.paranamer-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javax.transaction.transaction-api-LICENSE
apache-hive-3.1.2-bin/binary-package-licenses/javolution-LICENSE
```

- sudo mv apache-hive-3.1.2-bin hive
- sudo chmod 777 hive
- cd /home/hduser



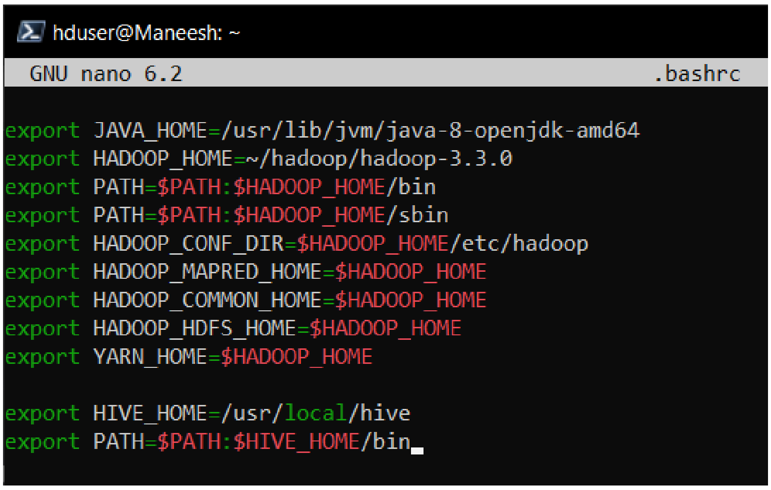
```
hduser@Maneesh: ~
hduser@Maneesh:/usr/local$ sudo mv apache-hive-3.1.2-bin hive
hduser@Maneesh:/usr/local$ sudo chmod 777 hive
hduser@Maneesh:/usr/local$ cd /home/hduser
hduser@Maneesh:~$ sudo nano .bashrc
```

5. In the sudo nano .bashrc file, add these lines very carefully:



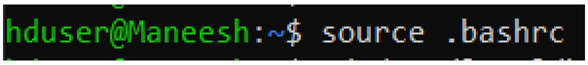
```
hduser@Maneesh:~$ sudo nano .bashrc
```

export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin



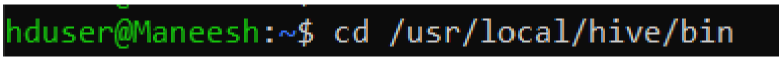
```
hduser@Maneesh: ~
  GNU nano 6.2                                        .bashrc

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=~/hadoop/hadoop-3.3.0
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME

export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
```

6. Then run command:
   - source .bashrc



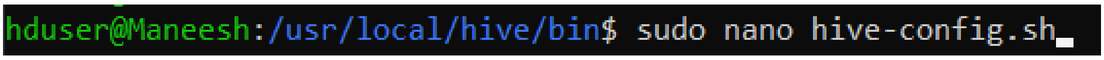
```
hduser@Maneesh:~$ source .bashrc
```

7. Then go to hive bin directory by command:
   - cd /usr/local/hive/bin



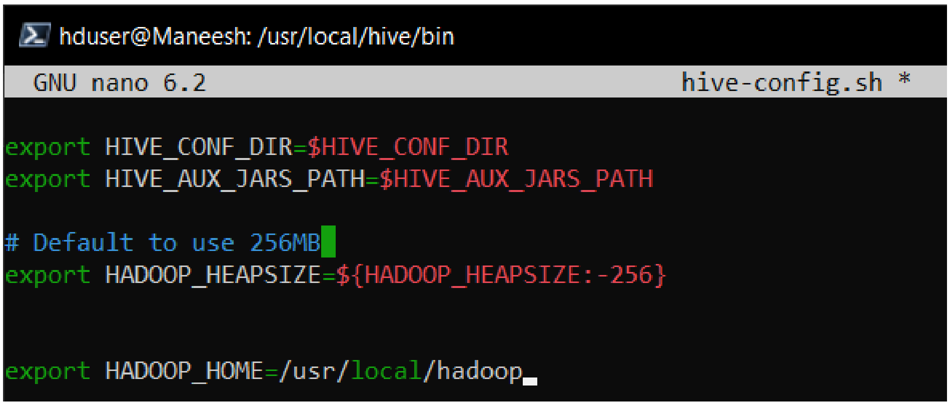
```
hduser@Maneesh:~$ cd /usr/local/hive/bin
```

8. Add the following line to sudo nano hive-config.sh



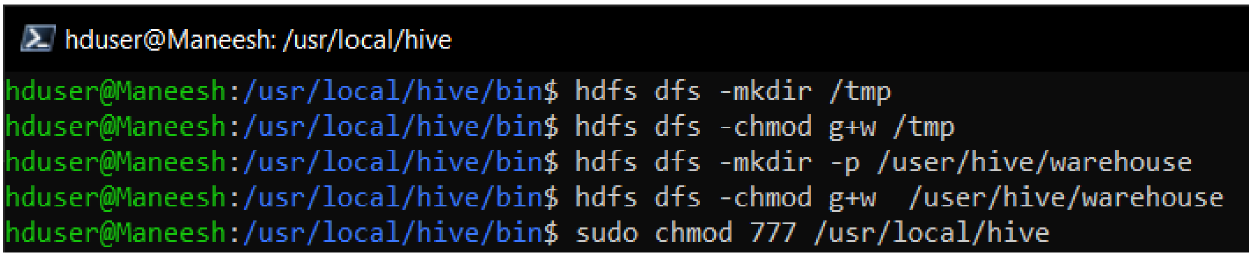
```
hduser@Maneesh:/usr/local/hive/bin$ sudo nano hive-config.sh
```

export HADOOP_HOME=/usr/local/hadoop



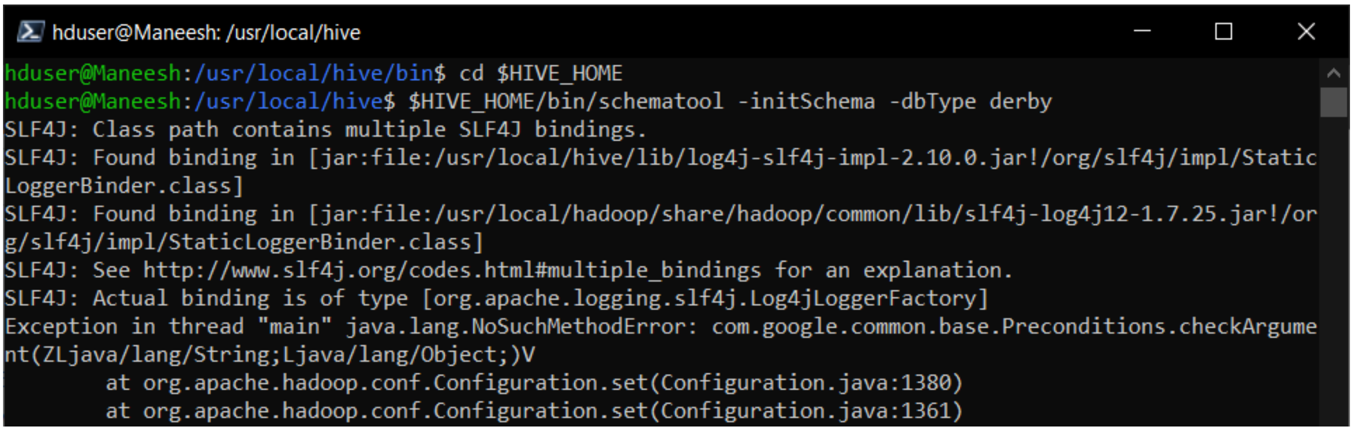
```
hduser@Maneesh: /usr/local/hive/bin
  GNU nano 6.2                                hive-config.sh *

export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH

# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}


export HADOOP_HOME=/usr/local/hadoop
```

9. Hive is installed now, but you need to first create some directories in HDFS for Hive to store its data.
   - ➢ hdfs dfs -mkdir /tmp
   - ➢ hdfs dfs -chmod g+w /tmp
   - ➢ hdfs dfs -mkdir -p /user/hive/warehouse
   - ➢ hdfs dfs -chmod g+w /user/hive/warehouse
   - ➢ sudo chmod 777 /usr/local/hive



```
hduser@Maneesh:/usr/local/hive/bin$ hdfs dfs -mkdir /tmp
hduser@Maneesh:/usr/local/hive/bin$ hdfs dfs -chmod g+w /tmp
hduser@Maneesh:/usr/local/hive/bin$ hdfs dfs -mkdir -p /user/hive/warehouse
hduser@Maneesh:/usr/local/hive/bin$ hdfs dfs -chmod g+w  /user/hive/warehouse
hduser@Maneesh:/usr/local/hive/bin$ sudo chmod 777 /usr/local/hive
```

10. Now we need to initialize derby database.
   - ➢ cd $HIVE_HOME
   - ➢ $HIVE_HOME/bin/schematool -initSchema -dbType derby

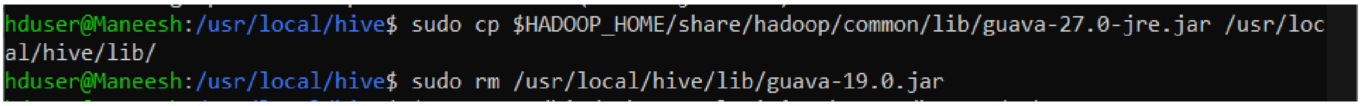


```
hduser@Maneesh:/usr/local/hive/bin$ cd $HIVE_HOME
hduser@Maneesh:/usr/local/hive$ $HIVE_HOME/bin/schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Exception in thread "main" java.lang.NoSuchMethodError: com.google.common.base.Preconditions.checkArgume
nt(ZLjava/lang/String;Ljava/lang/Object;)V
        at org.apache.hadoop.conf.Configuration.set(Configuration.java:1380)
        at org.apache.hadoop.conf.Configuration.set(Configuration.java:1361)
```

You may face error like ‘NoSuchMethodFound’.

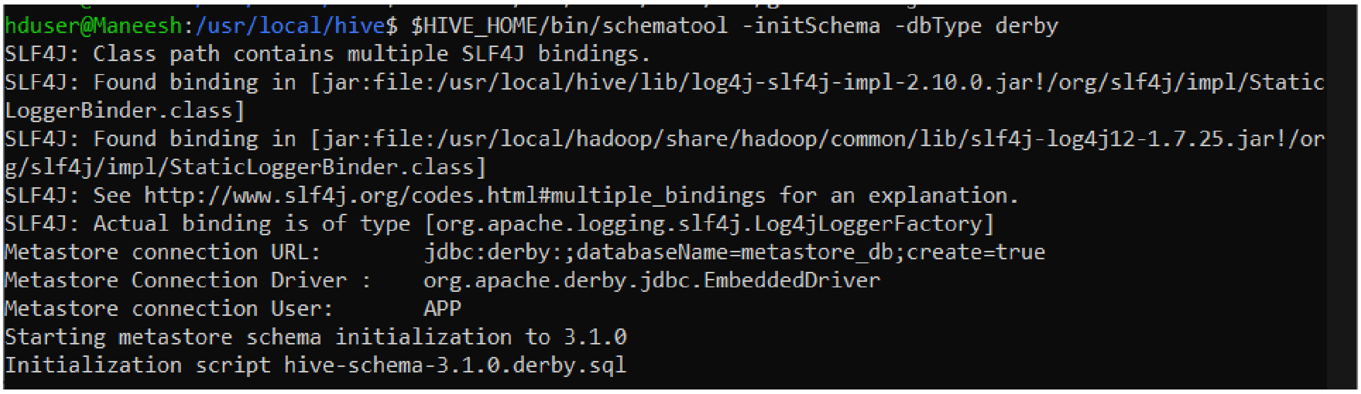
To solve it:

- ➢ sudo cp $HADOOP_HOME/share/hadoop/common/lib/guava-27.0-jre.jar /usr/local/hive/lib/
- ➢ sudo rm /usr/local/hive/lib/guava-19.0.jar



```
hduser@Maneesh:/usr/local/hive$ sudo cp $HADOOP_HOME/share/hadoop/common/lib/guava-27.0-jre.jar /usr/loc
al/hive/lib/
hduser@Maneesh:/usr/local/hive$ sudo rm /usr/local/hive/lib/guava-19.0.jar
```

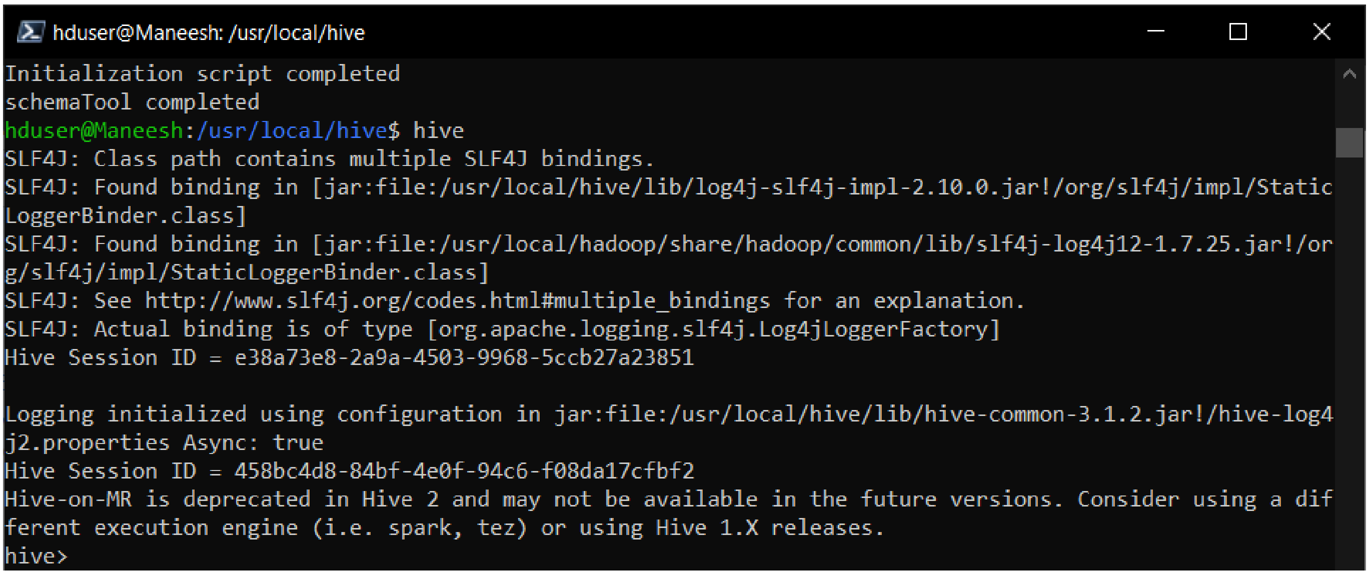
- ➢ cd $HIVE_HOME
- ➢ $HIVE_HOME/bin/schematool -initSchema -dbType derby



```
hduser@Maneesh:/usr/local/hive$ $HIVE_HOME/bin/schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

Start hive shell by typing:

- hive

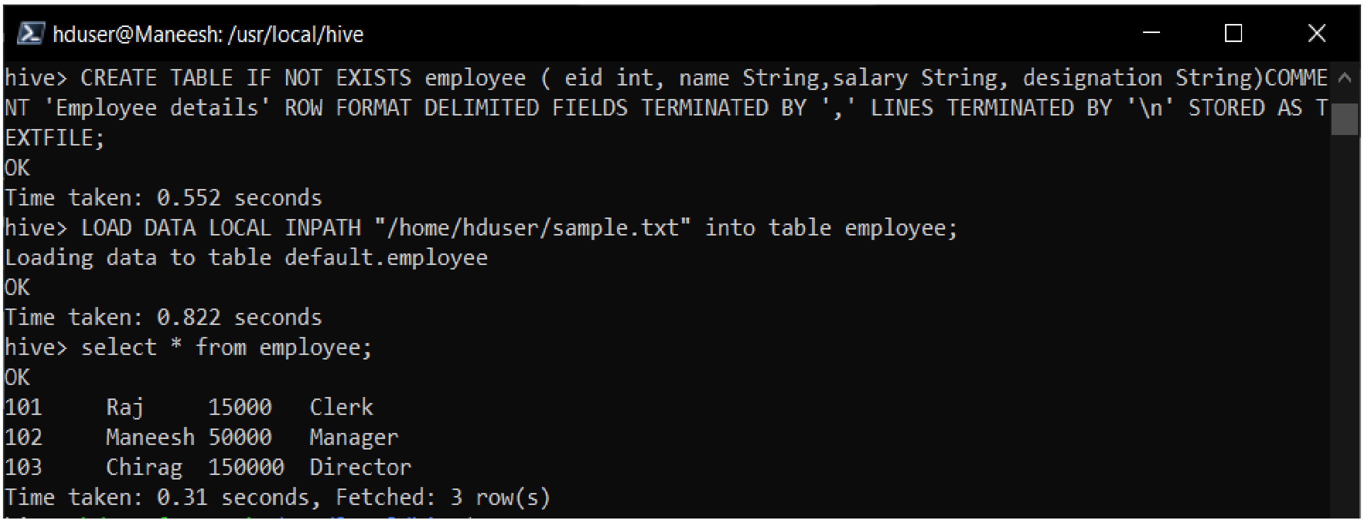


```
Initialization script completed
schemaTool completed
hduser@Maneesh:/usr/local/hive$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = e38a73e8-2a9a-4503-9968-5ccb27a23851

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-3.1.2.jar!/hive-log4
j2.properties Async: true
Hive Session ID = 458bc4d8-84bf-4e0f-94c6-f08da17cfbf2
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a dif
ferent execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive>
```

- CREATE TABLE IF NOT EXISTS employee (eid int, name String,salary String, designation String)COMMENT 'Employee details' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

- LOAD DATA LOCAL INPATH "/home/hduser/sample.txt" into table employee;

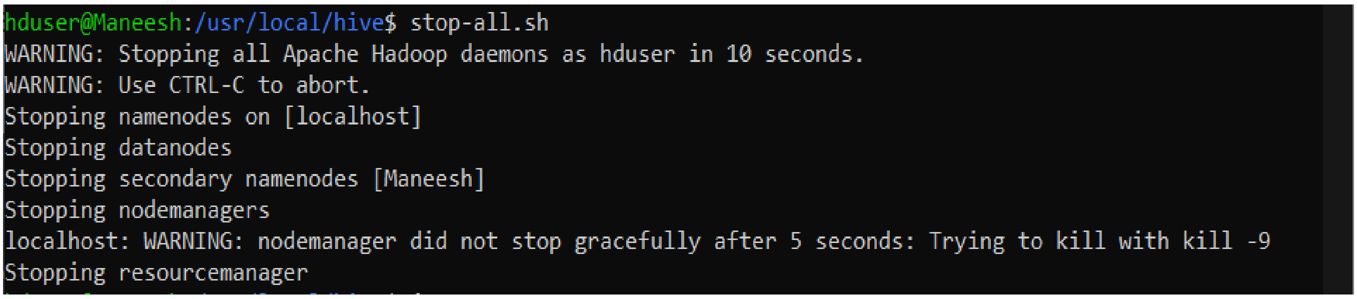
- select * from employee;



```
hive> CREATE TABLE IF NOT EXISTS employee ( eid int, name String,salary String, designation String)COMME
NT 'Employee details' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS T
EXTFILE;
OK
Time taken: 0.552 seconds
hive> LOAD DATA LOCAL INPATH "/home/hduser/sample.txt" into table employee;
Loading data to table default.employee
OK
Time taken: 0.822 seconds
hive> select * from employee;
OK
101     Raj     15000   Clerk
102     Maneesh 50000   Manager
103     Chirag  150000  Director
Time taken: 0.31 seconds, Fetched: 3 row(s)
```

Then stop hadoop and close.

- stop-all.sh



```
hduser@Maneesh:/usr/local/hive$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [Maneesh]
Stopping nodemanagers
localhost: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill with kill -9
Stopping resourcemanager
```

Practical No: 5

Aim: Install HBase and use the HBase Data model Store and retrieve data.

Theory:

HBase is a column-oriented non-relational database management system that runs on top of Hadoop Distributed File System (HDFS). HBase provides a fault-tolerant way of storing sparse data sets, which are common in many big data use cases. It is well suited for real-time data processing or random read/write access to large volumes of data.

Unlike relational database systems, HBase does not support a structured query language like SQL; in fact, HBase isn't a relational data store at all. HBase applications are written in Java™ much like a typical Apache MapReduce application. HBase does support writing applications in Apache Avro, REST and Thrift.

An HBase system is designed to scale linearly. It comprises a set of standard tables with rows and columns, much like a traditional database. Each table must have an element defined as a primary key, and all access attempts to HBase tables must use this primary key.

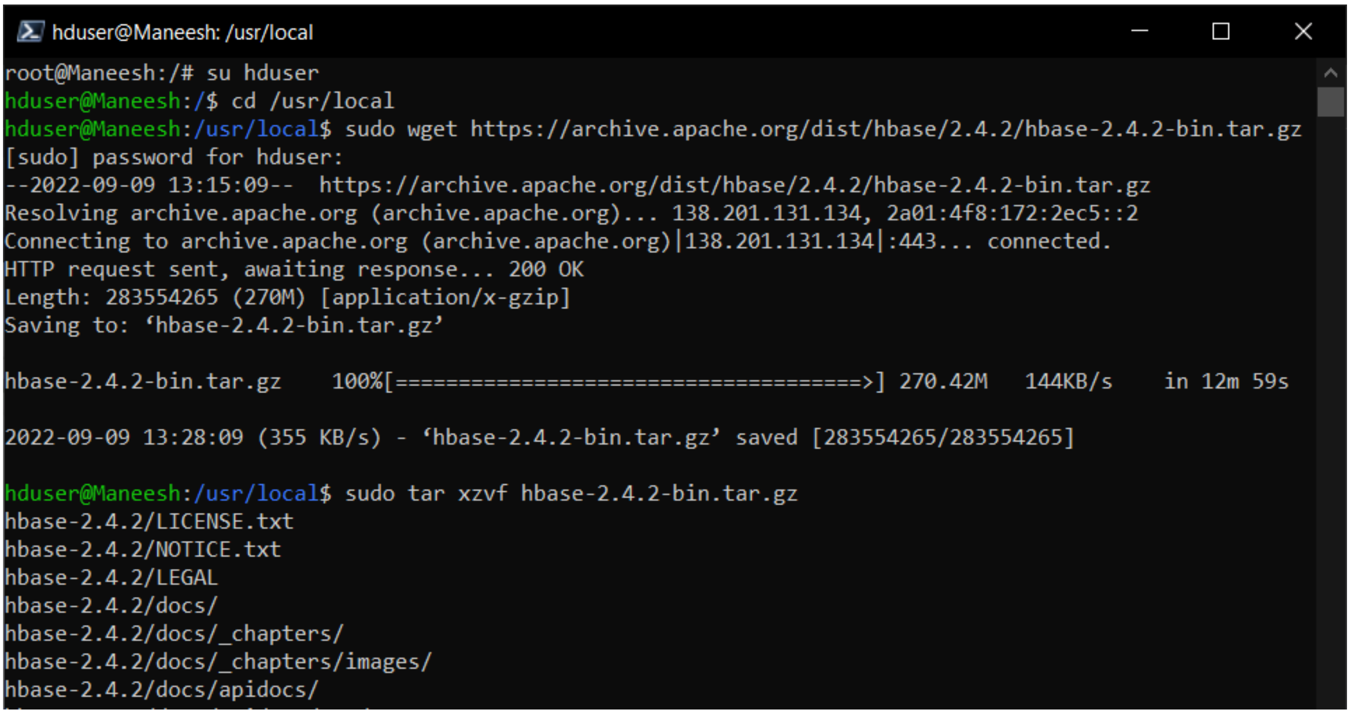
Avro, as a component, supports a rich set of primitive data types including: numeric, binary data and strings; and a number of complex types including arrays, maps, enumerations and records. A sort order can also be defined for the data.

HBase relies on ZooKeeper for high-performance coordination. ZooKeeper is built into HBase, but if you're running a production cluster, it's suggested that you have a dedicated ZooKeeper cluster that's integrated with your HBase cluster.

HBase works well with Hive, a query engine for batch processing of big data, to enable fault-tolerant big data applications.

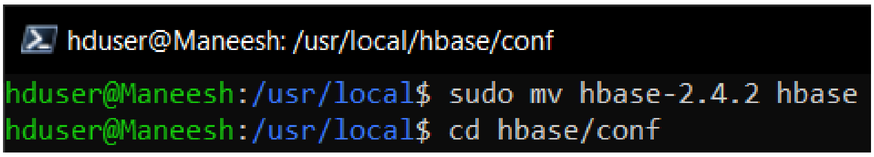
Steps:

1. Installation Part:
   - su hduser
   - cd /usr/local
   - sudo wget https://archive.apache.org/dist/hbase/2.4.2/hbase-2.4.2-bin.tar.gz
   - sudo tar xzvf hbase-2.4.2-bin.tar.gz



```
root@Maneesh:/# su hduser
hduser@Maneesh:/$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo wget https://archive.apache.org/dist/hbase/2.4.2/hbase-2.4.2-bin.tar.gz
[sudo] password for hduser:
--2022-09-09 13:15:09--  https://archive.apache.org/dist/hbase/2.4.2/hbase-2.4.2-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 283554265 (270M) [application/x-gzip]
Saving to: 'hbase-2.4.2-bin.tar.gz'

hbase-2.4.2-bin.tar.gz    100%[=====================================>] 270.42M   144KB/s    in 12m 59s

2022-09-09 13:28:09 (355 KB/s) - 'hbase-2.4.2-bin.tar.gz' saved [283554265/283554265]

hduser@Maneesh:/usr/local$ sudo tar xzvf hbase-2.4.2-bin.tar.gz
hbase-2.4.2/LICENSE.txt
hbase-2.4.2/NOTICE.txt
hbase-2.4.2/LEGAL
hbase-2.4.2/docs/
hbase-2.4.2/docs/_chapters/
hbase-2.4.2/docs/_chapters/images/
hbase-2.4.2/docs/apidocs/
```

- sudo mv hbase-2.4.2 hbase
- cd hbase/conf

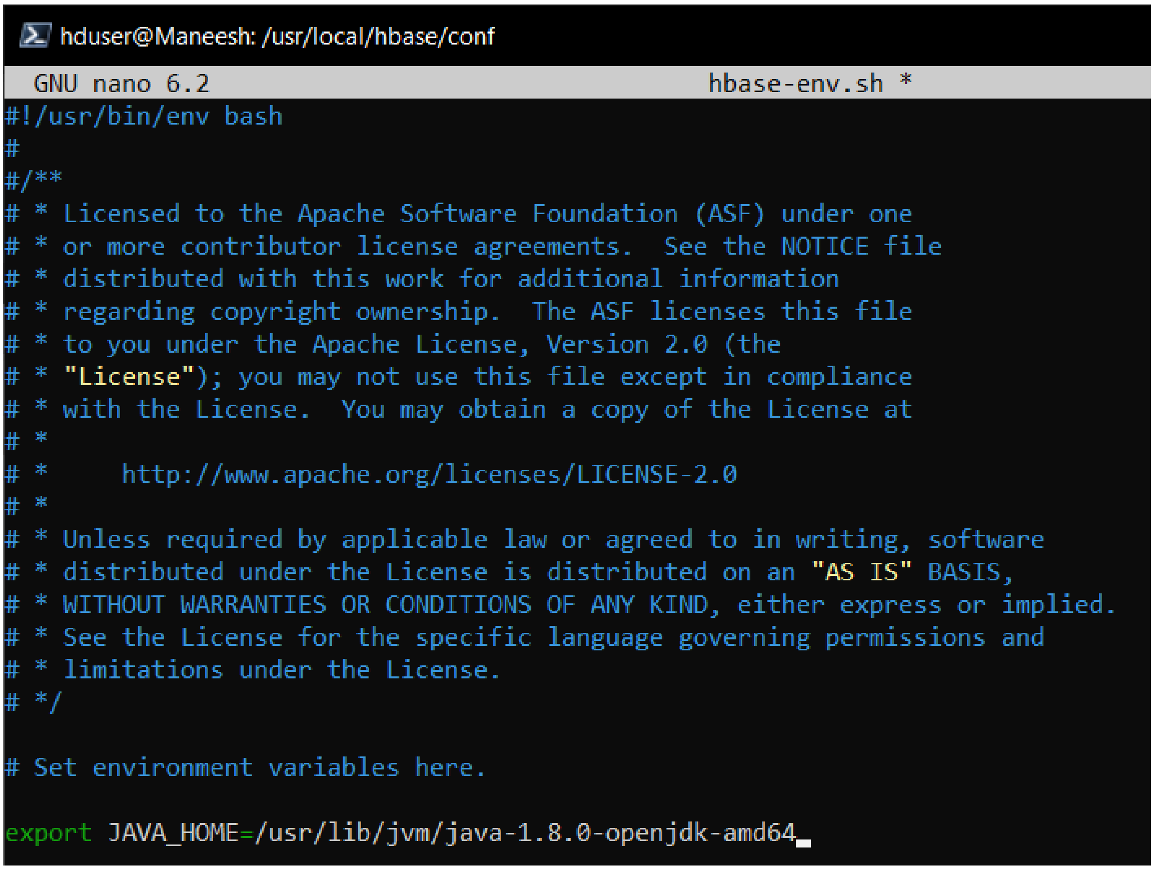


```
hduser@Maneesh: /usr/local/hbase/conf
hduser@Maneesh:/usr/local$ sudo mv hbase-2.4.2 hbase
hduser@Maneesh:/usr/local$ cd hbase/conf
```

2. Add the following line to sudo nano hbase-env.sh:

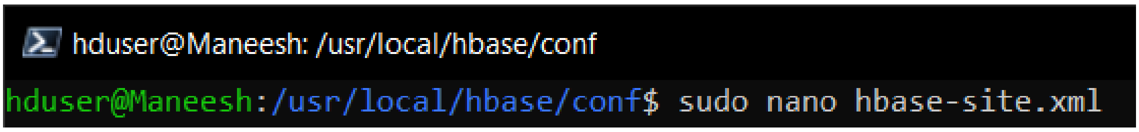
```
hduser@Maneesh:/usr/local/hbase/conf$ sudo nano hbase-env.sh
```

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64



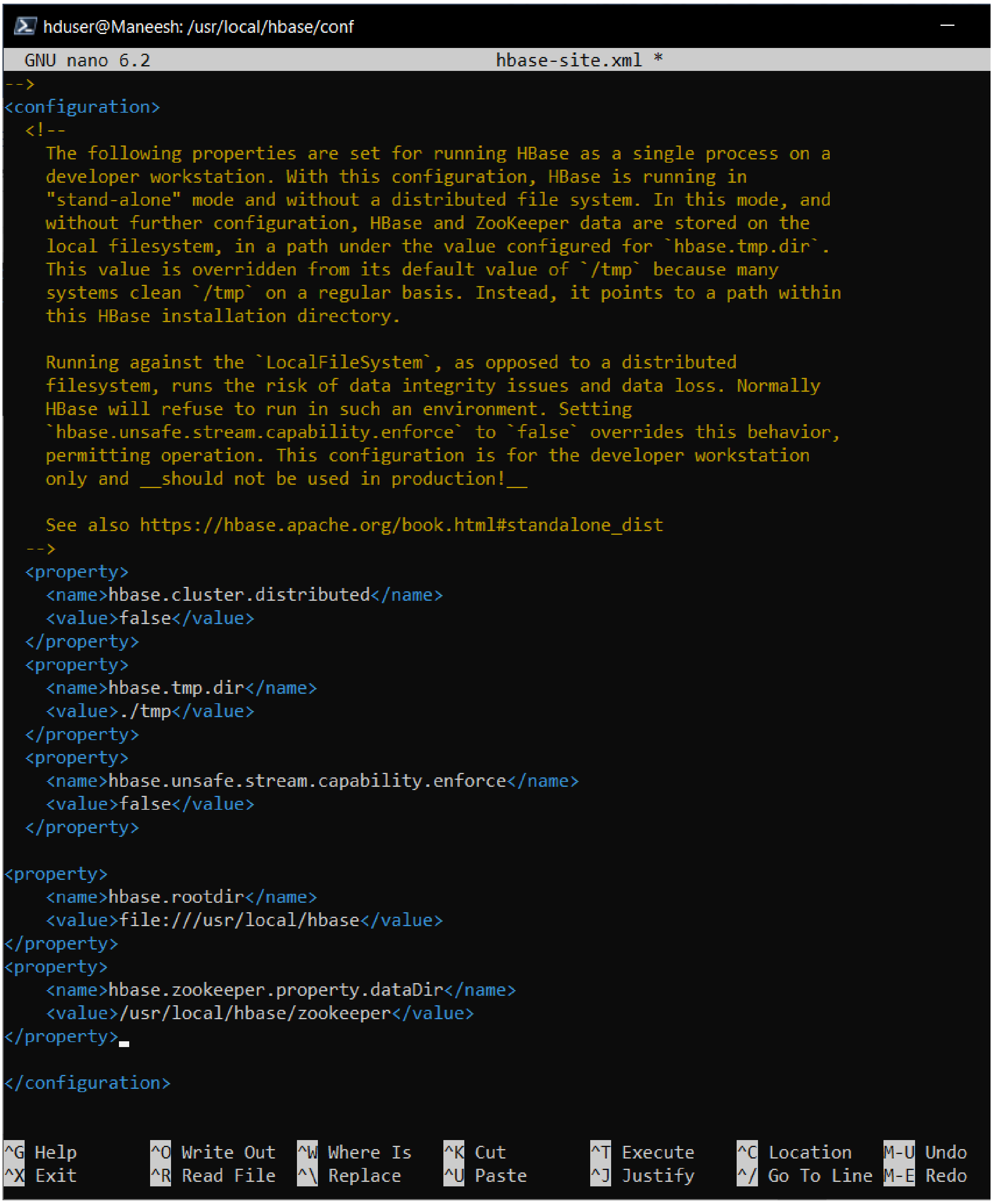
```
hduser@Maneesh: /usr/local/hbase/conf
  GNU nano 6.2                                hbase-env.sh *
#!/usr/bin/env bash
#
#/**
# * Licensed to the Apache Software Foundation (ASF) under one
# * or more contributor license agreements.  See the NOTICE file
# * distributed with this work for additional information
# * regarding copyright ownership.  The ASF licenses this file
# * to you 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.
# */

# Set environment variables here.

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

3. Add the following lines between <configuration> and </configuration> of sudo nano hbase-site.xml:



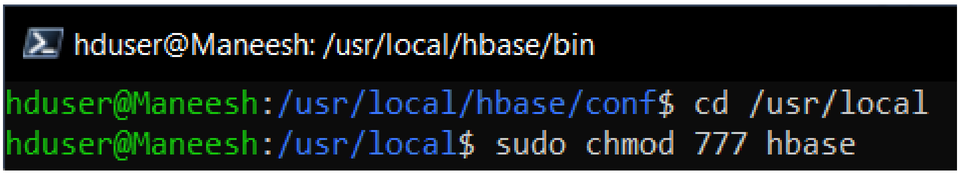
```
hduser@Maneesh: /usr/local/hbase/conf
hduser@Maneesh:/usr/local/hbase/conf$ sudo nano hbase-site.xml
```

```
<property>
   <name>hbase.rootdir</name>
   <value>file:///usr/local/hbase</value>
</property>
<property>
   <name>hbase.zookeeper.property.dataDir</name>
   <value>/usr/local/hbase/zookeeper</value>
</property>
```



```
hduser@Maneesh: /usr/local/hbase/conf
  GNU nano 6.2                            hbase-site.xml *
-->
<configuration>
  <!--
    The following properties are set for running HBase as a single process on a
    developer workstation. With this configuration, HBase is running in
    "stand-alone" mode and without a distributed file system. In this mode, and
    without further configuration, HBase and ZooKeeper data are stored on the
    local filesystem, in a path under the value configured for `hbase.tmp.dir`.
    This value is overridden from its default value of `/tmp` because many
    systems clean `/tmp` on a regular basis. Instead, it points to a path within
    this HBase installation directory.

    Running against the `LocalFileSystem`, as opposed to a distributed
    filesystem, runs the risk of data integrity issues and data loss. Normally
    HBase will refuse to run in such an environment. Setting
    `hbase.unsafe.stream.capability.enforce` to `false` overrides this behavior,
    permitting operation. This configuration is for the developer workstation
    only and __should not be used in production!__

    See also https://hbase.apache.org/book.html#standalone_dist
  -->
  <property>
    <name>hbase.cluster.distributed</name>
    <value>false</value>
  </property>
  <property>
    <name>hbase.tmp.dir</name>
    <value>./tmp</value>
  </property>
  <property>
    <name>hbase.unsafe.stream.capability.enforce</name>
    <value>false</value>
  </property>

<property>
    <name>hbase.rootdir</name>
    <value>file:///usr/local/hbase</value>
</property>
<property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/usr/local/hbase/zookeeper</value>
</property>

</configuration>

^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
```

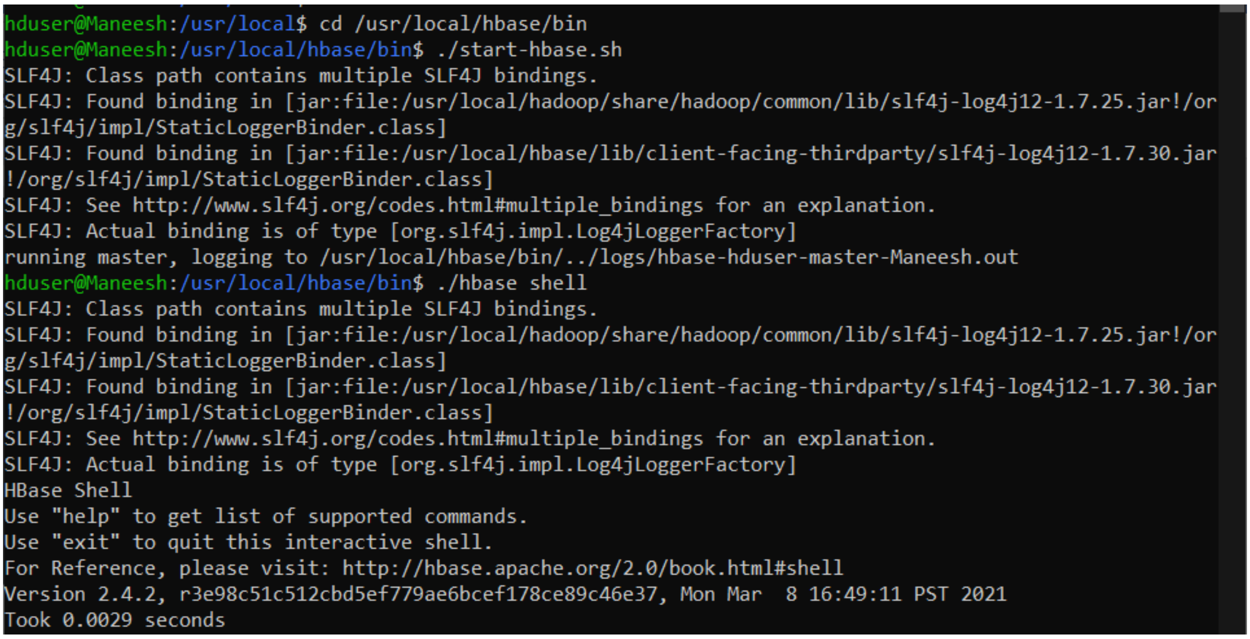
4. Give permission to hbase folder.
   - cd /usr/local
   - sudo chmod 777 hbase



```
hduser@Maneesh: /usr/local/hbase/bin
hduser@Maneesh:/usr/local/hbase/conf$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo chmod 777 hbase
```

Now to start HBase and insert data:

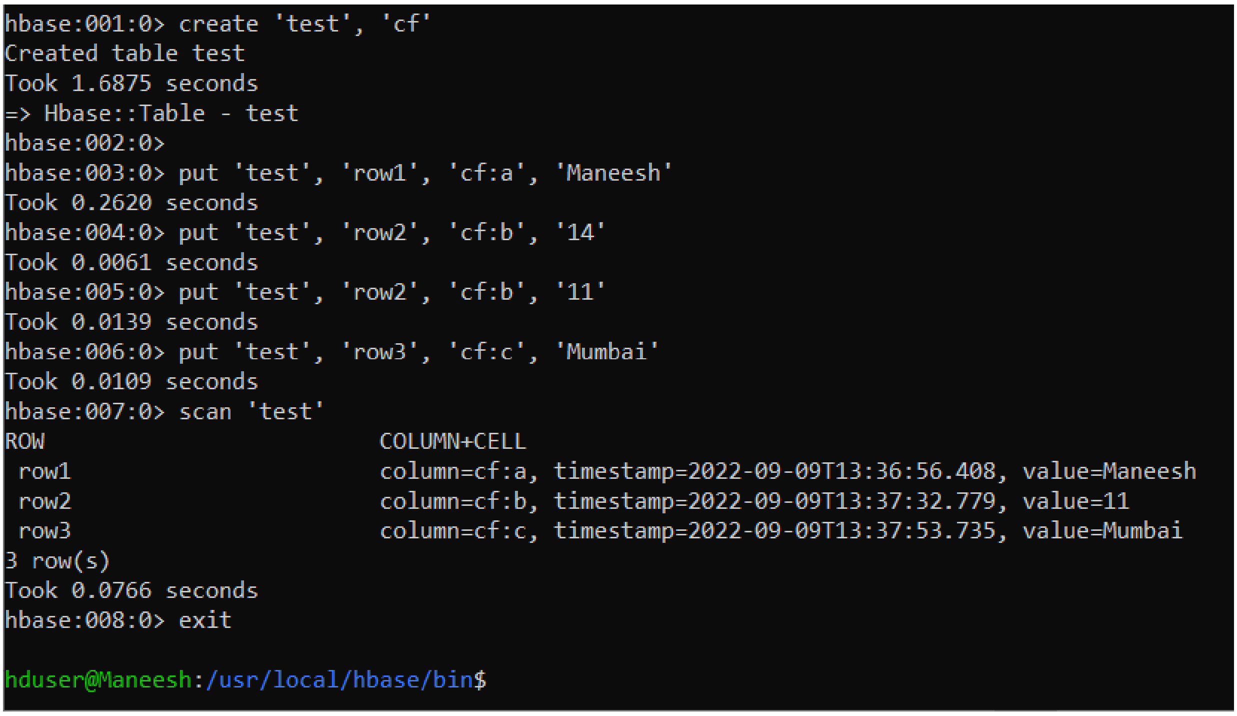
- cd /usr/local/hbase/bin
- ./start-hbase.sh
- ./hbase shell



```
hduser@Maneesh:/usr/local$ cd /usr/local/hbase/bin
hduser@Maneesh:/usr/local/hbase/bin$ ./start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
running master, logging to /usr/local/hbase/bin/../logs/hbase-hduser-master-Maneesh.out
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.2, r3e98c51c512cbd5ef779ae6bcef178ce89c46e37, Mon Mar  8 16:49:11 PST 2021
Took 0.0029 seconds
```

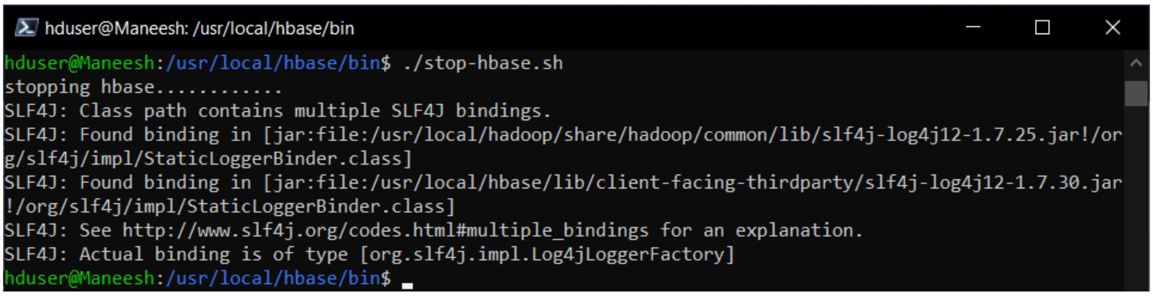
Inside the hbase shell:

```
create 'test', 'cf'
put 'test', 'row1', 'cf:a', 'Maneesh'
put 'test', 'row2', 'cf:b', '11'
put 'test', 'row3', 'cf:c', 'Mumbai'
scan 'test'
exit
```



```
hbase:001:0> create 'test', 'cf'
Created table test
Took 1.6875 seconds
=> Hbase::Table - test
hbase:002:0>
hbase:003:0> put 'test', 'row1', 'cf:a', 'Maneesh'
Took 0.2620 seconds
hbase:004:0> put 'test', 'row2', 'cf:b', '14'
Took 0.0061 seconds
hbase:005:0> put 'test', 'row2', 'cf:b', '11'
Took 0.0139 seconds
hbase:006:0> put 'test', 'row3', 'cf:c', 'Mumbai'
Took 0.0109 seconds
hbase:007:0> scan 'test'
ROW                          COLUMN+CELL
 row1                        column=cf:a, timestamp=2022-09-09T13:36:56.408, value=Maneesh
 row2                        column=cf:b, timestamp=2022-09-09T13:37:32.779, value=11
 row3                        column=cf:c, timestamp=2022-09-09T13:37:53.735, value=Mumbai
3 row(s)
Took 0.0766 seconds
hbase:008:0> exit

hduser@Maneesh:/usr/local/hbase/bin$
```

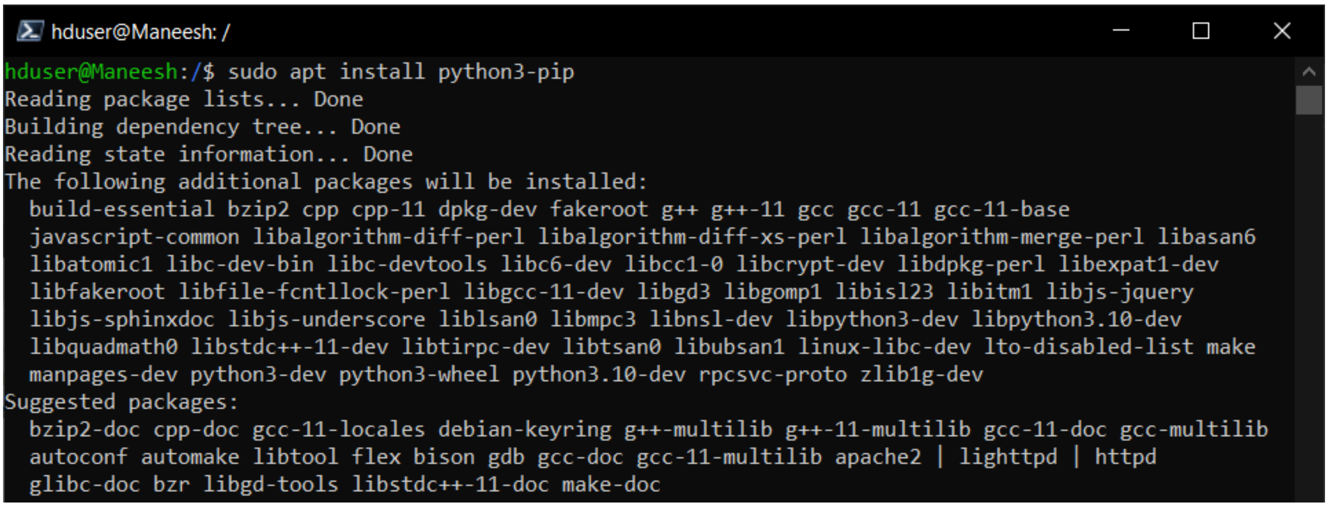
- ➢ ./stop-hbase.sh



```
hduser@Maneesh:/usr/local/hbase/bin$ ./stop-hbase.sh
stopping hbase............
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hduser@Maneesh:/usr/local/hbase/bin$ _
```

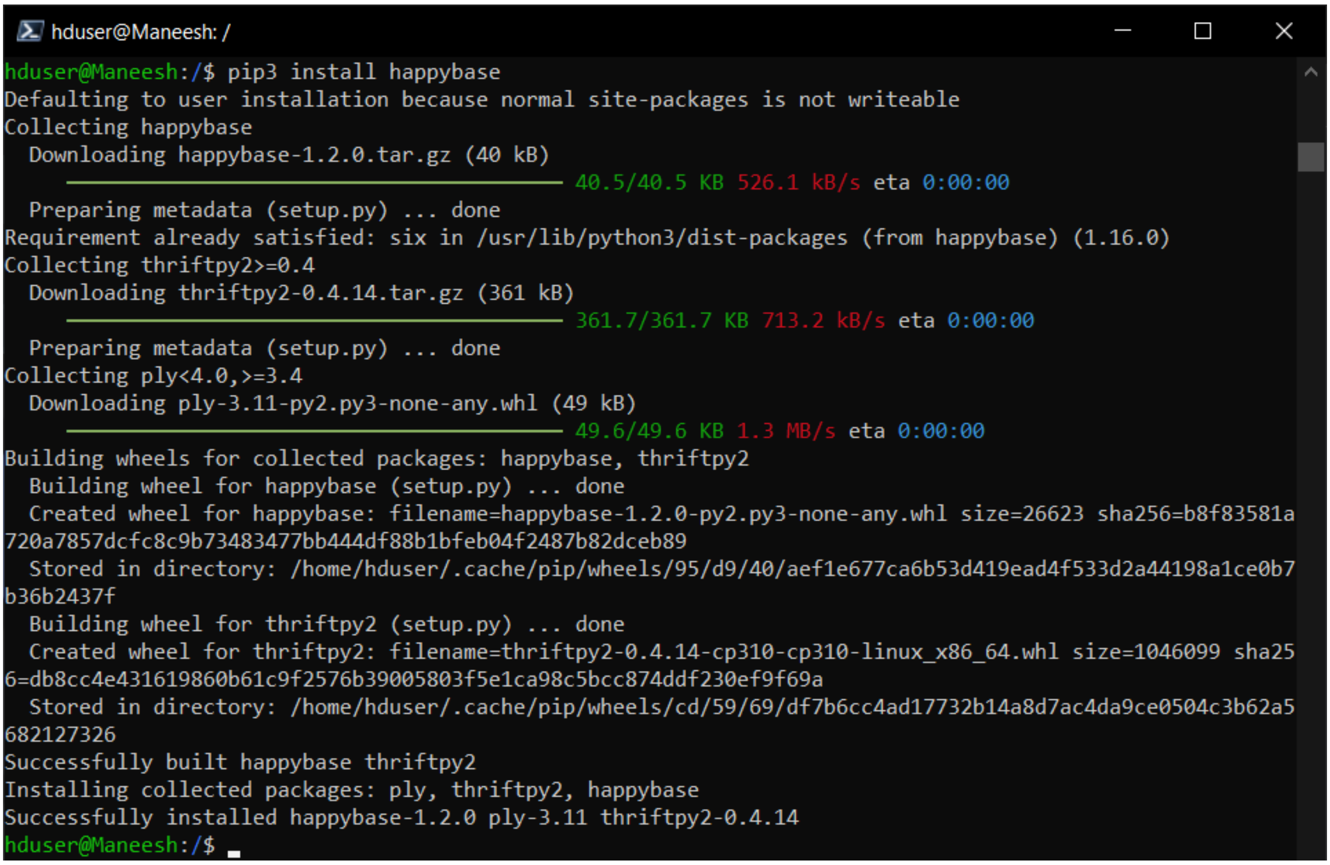
Now we shall access this data using Python:

- sudo apt install python3-pip



```
hduser@Maneesh:/$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot g++ g++-11 gcc gcc-11 gcc-11-base
  javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6
  libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libdpkg-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjs-jquery
  libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3-dev libpython3.10-dev
  libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list make
  manpages-dev python3-dev python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  bzip2-doc cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib
  autoconf automake libtool flex bison gdb gcc-doc gcc-11-multilib apache2 | lighttpd | httpd
  glibc-doc bzr libgd-tools libstdc++-11-doc make-doc
```

- pip3 install happybase

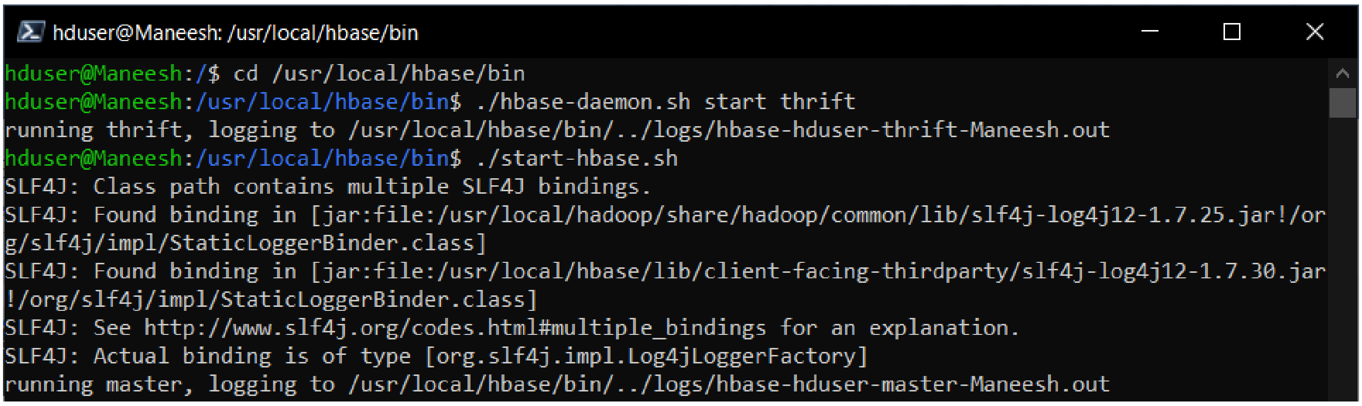


```
hduser@Maneesh:/$ pip3 install happybase
Defaulting to user installation because normal site-packages is not writeable
Collecting happybase
  Downloading happybase-1.2.0.tar.gz (40 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40.5/40.5 KB 526.1 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from happybase) (1.16.0)
Collecting thriftpy2>=0.4
  Downloading thriftpy2-0.4.14.tar.gz (361 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 361.7/361.7 KB 713.2 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting ply<4.0,>=3.4
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 KB 1.3 MB/s eta 0:00:00
Building wheels for collected packages: happybase, thriftpy2
  Building wheel for happybase (setup.py) ... done
  Created wheel for happybase: filename=happybase-1.2.0-py2.py3-none-any.whl size=26623 sha256=b8f83581a
720a7857dcfc8c9b73483477bb444df88b1bfeb04f2487b82dceb89
  Stored in directory: /home/hduser/.cache/pip/wheels/95/d9/40/aef1e677ca6b53d419ead4f533d2a44198a1ce0b7
b36b2437f
  Building wheel for thriftpy2 (setup.py) ... done
  Created wheel for thriftpy2: filename=thriftpy2-0.4.14-cp310-cp310-linux_x86_64.whl size=1046099 sha25
6=db8cc4e431619860b61c9f2576b39005803f5e1ca98c5bcc874ddf230ef9f69a
  Stored in directory: /home/hduser/.cache/pip/wheels/cd/59/69/df7b6cc4ad17732b14a8d7ac4da9ce0504c3b62a5
682127326
Successfully built happybase thriftpy2
Installing collected packages: ply, thriftpy2, happybase
Successfully installed happybase-1.2.0 ply-3.11 thriftpy2-0.4.14
hduser@Maneesh:/$ _
```

Python data manipulation part:

(make sure that you start the thrift server first and then the HBase, and while closing you stop HBase first and then thrift server)

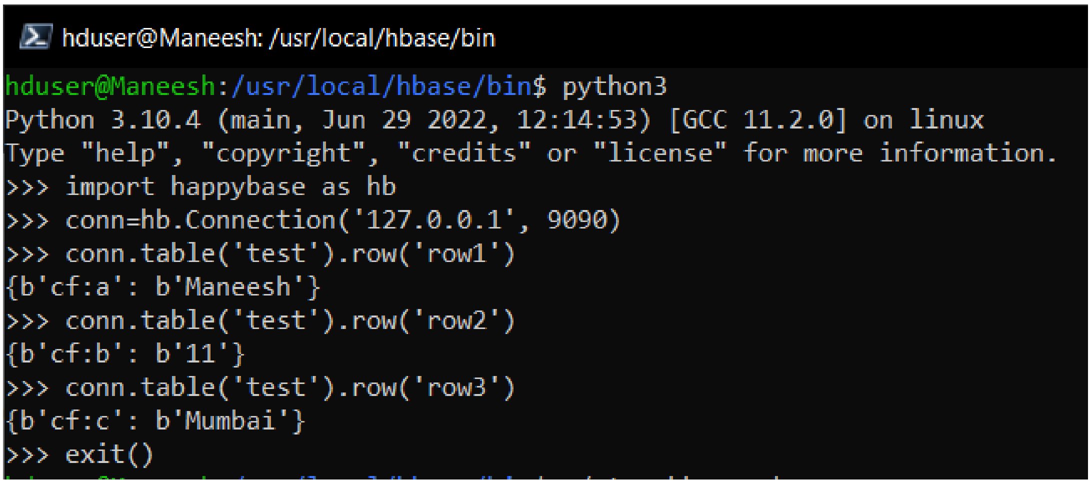
- cd /usr/local/hbase/bin
- ./hbase-daemon.sh start thrift
- ./start-hbase.sh



```
hduser@Maneesh:/$ cd /usr/local/hbase/bin
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase-daemon.sh start thrift
running thrift, logging to /usr/local/hbase/bin/../logs/hbase-hduser-thrift-Maneesh.out
hduser@Maneesh:/usr/local/hbase/bin$ ./start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
running master, logging to /usr/local/hbase/bin/../logs/hbase-hduser-master-Maneesh.out
```

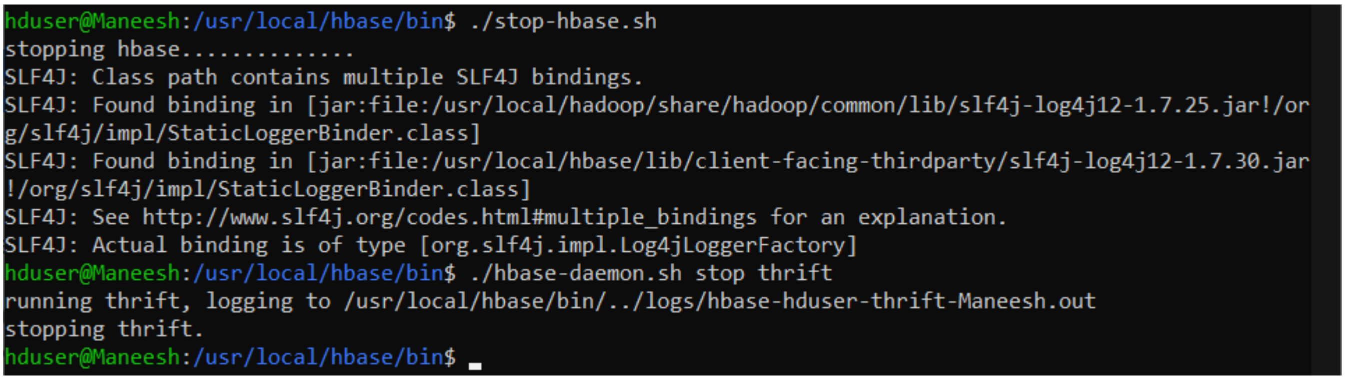
- python3

  import happybase as hb
  conn=hb.Connection('127.0.0.1', 9090)
  conn.table('test').row('row1')
  conn.table('test').row('row2')
  conn.table('test').row('row3')
  exit()



```
hduser@Maneesh:/usr/local/hbase/bin$ python3
Python 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import happybase as hb
>>> conn=hb.Connection('127.0.0.1', 9090)
>>> conn.table('test').row('row1')
{b'cf:a': b'Maneesh'}
>>> conn.table('test').row('row2')
{b'cf:b': b'11'}
>>> conn.table('test').row('row3')
{b'cf:c': b'Mumbai'}
>>> exit()
```

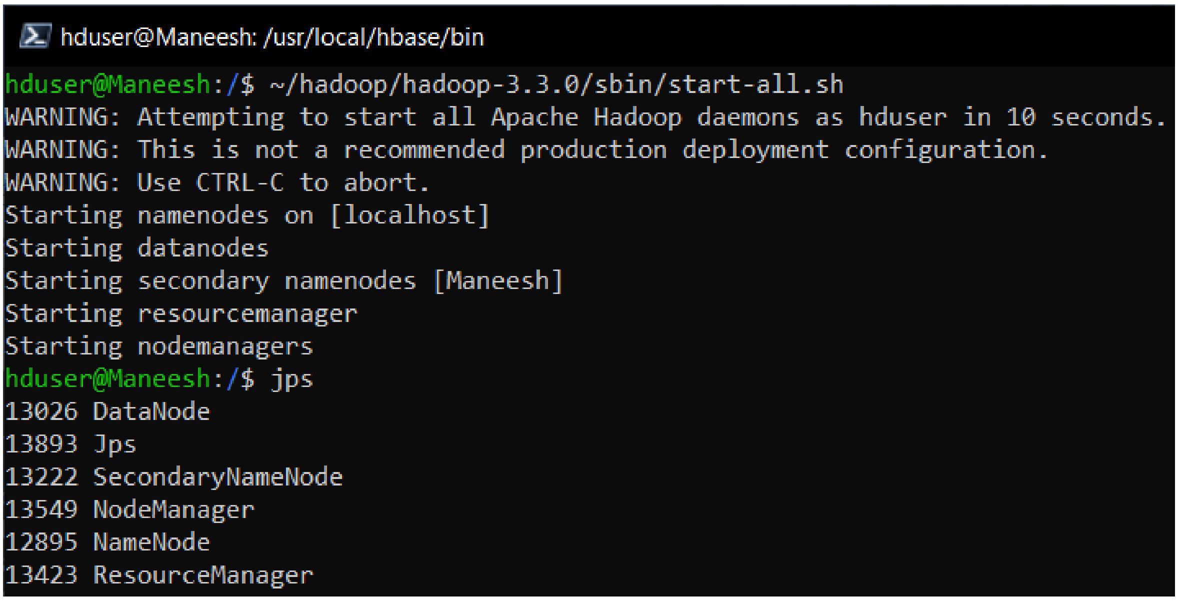
- ./stop-hbase.sh
- ./hbase-daemon.sh stop thrift



```
hduser@Maneesh:/usr/local/hbase/bin$ ./stop-hbase.sh
stopping hbase.............
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase-daemon.sh stop thrift
running thrift, logging to /usr/local/hbase/bin/../logs/hbase-hduser-thrift-Maneesh.out
stopping thrift.
hduser@Maneesh:/usr/local/hbase/bin$ _
```

HDFS to HBASE:

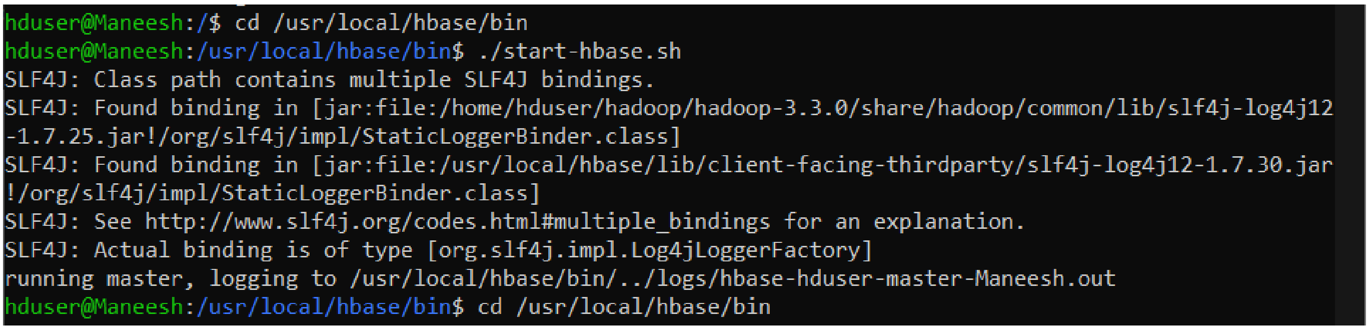
1. Start Hadoop and HBase
   - sudo service ssh start
   - ssh localhost
   - /usr/local/hadoop/sbin/start-all.sh
   - jps



hduser@Maneesh: /usr/local/hbase/bin

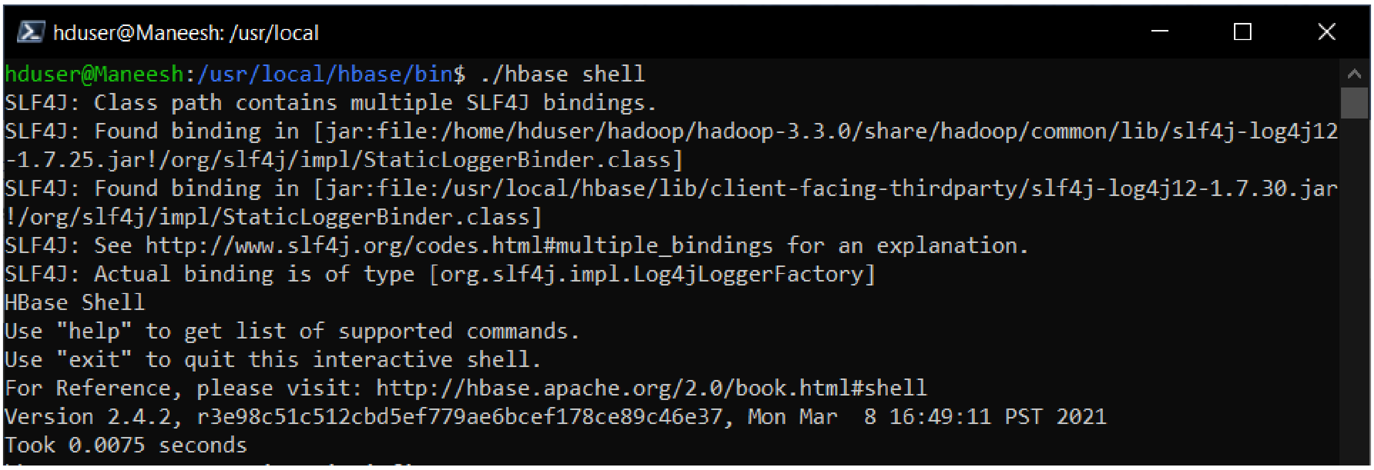
```
hduser@Maneesh:/$ ~/hadoop/hadoop-3.3.0/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Maneesh]
Starting resourcemanager
Starting nodemanagers
hduser@Maneesh:/$ jps
13026 DataNode
13893 Jps
13222 SecondaryNameNode
13549 NodeManager
12895 NameNode
13423 ResourceManager
```

- ➢ cd /usr/local/hbase/bin
- ➢ ./start-hbase.sh
- ➢ cd /usr/local/hbase/bin



```
hduser@Maneesh:/$ cd /usr/local/hbase/bin
hduser@Maneesh:/usr/local/hbase/bin$ ./start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/hadoop-3.3.0/share/hadoop/common/lib/slf4j-log4j12
-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
running master, logging to /usr/local/hbase/bin/../logs/hbase-hduser-master-Maneesh.out
hduser@Maneesh:/usr/local/hbase/bin$ cd /usr/local/hbase/bin
```

2. Start HBase Shell
   - ➢ ./hbase shell

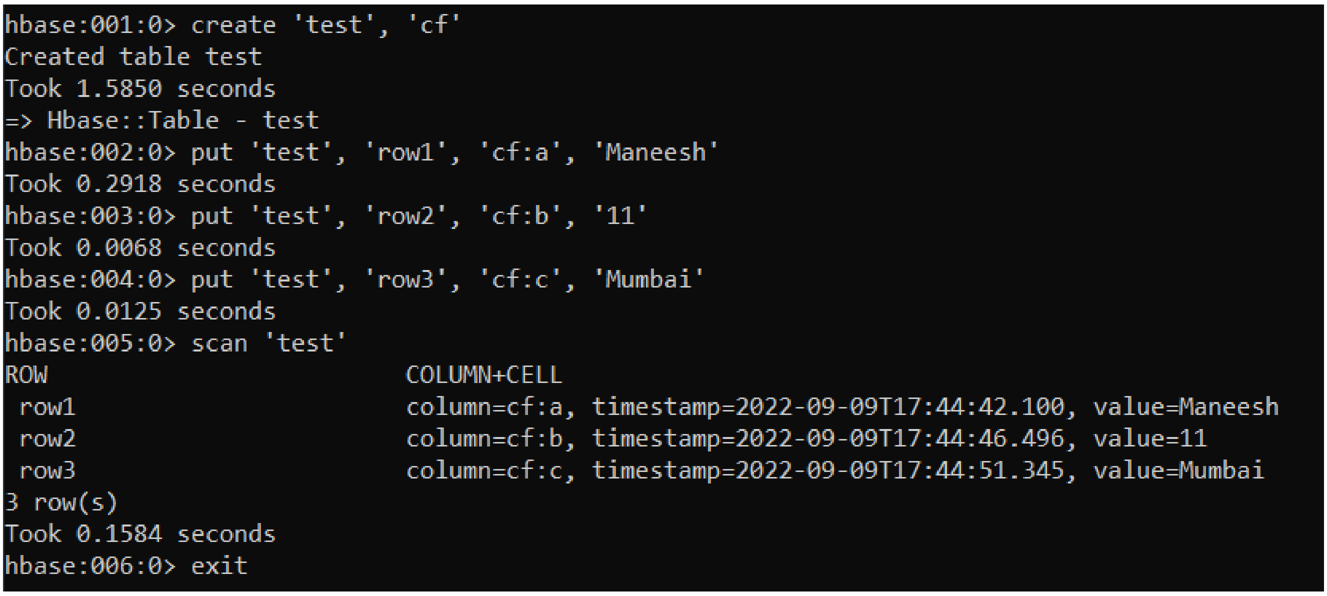


hduser@Maneesh: /usr/local

```
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/hadoop-3.3.0/share/hadoop/common/lib/slf4j-log4j12
-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.2, r3e98c51c512cbd5ef779ae6bcef178ce89c46e37, Mon Mar  8 16:49:11 PST 2021
Took 0.0075 seconds
```

3. Create a table in HBase:

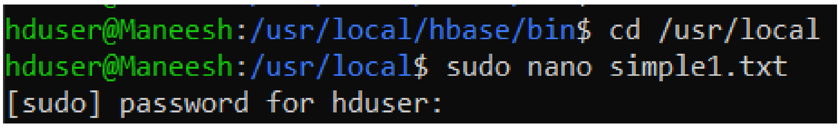
```
create 'test', 'cf'
put 'test', 'row1', 'cf:a', 'Maneesh'
put 'test', 'row2', 'cf:b', '11'
put 'test', 'row3', 'cf:c', 'Mumbai'
scan 'test'
exit
```



```
hbase:001:0> create 'test', 'cf'
Created table test
Took 1.5850 seconds
=> Hbase::Table - test
hbase:002:0> put 'test', 'row1', 'cf:a', 'Maneesh'
Took 0.2918 seconds
hbase:003:0> put 'test', 'row2', 'cf:b', '11'
Took 0.0068 seconds
hbase:004:0> put 'test', 'row3', 'cf:c', 'Mumbai'
Took 0.0125 seconds
hbase:005:0> scan 'test'
ROW                         COLUMN+CELL
 row1                       column=cf:a, timestamp=2022-09-09T17:44:42.100, value=Maneesh
 row2                       column=cf:b, timestamp=2022-09-09T17:44:46.496, value=11
 row3                       column=cf:c, timestamp=2022-09-09T17:44:51.345, value=Mumbai
3 row(s)
Took 0.1584 seconds
hbase:006:0> exit
```

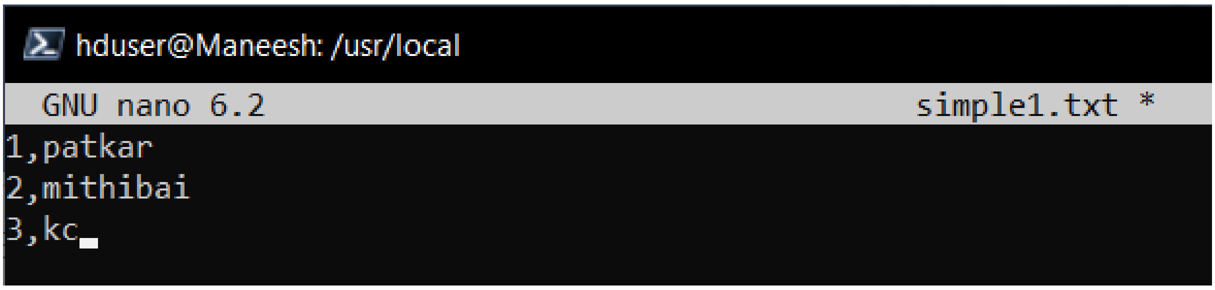
TO EXPORT TO HBASE

1. Create a text file in /usr/local named simple1.txt with the following contents:
   - cd /usr/local
   - nano simple1.txt



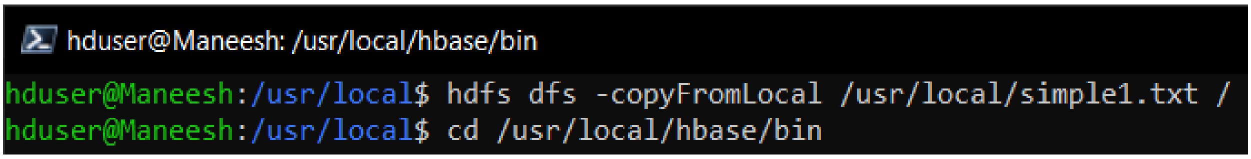
```
hduser@Maneesh:/usr/local/hbase/bin$ cd /usr/local
hduser@Maneesh:/usr/local$ sudo nano simple1.txt
[sudo] password for hduser:
```

1,patkar
2,mithibai
3,kc



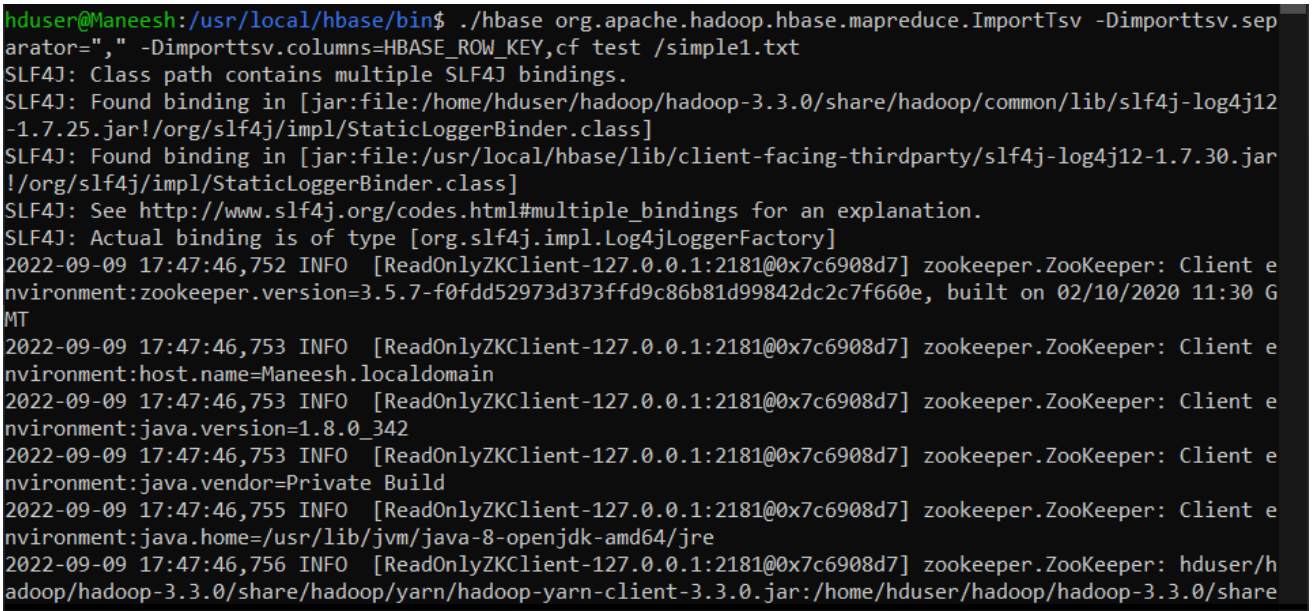
```
  GNU nano 6.2                                  simple1.txt *
1,patkar
2,mithibai
3,kc
```

2. Copy file to HDFS
   - hdfs dfs -copyFromLocal /usr/local/simple1.txt /
   - cd /usr/local/hbase/bin

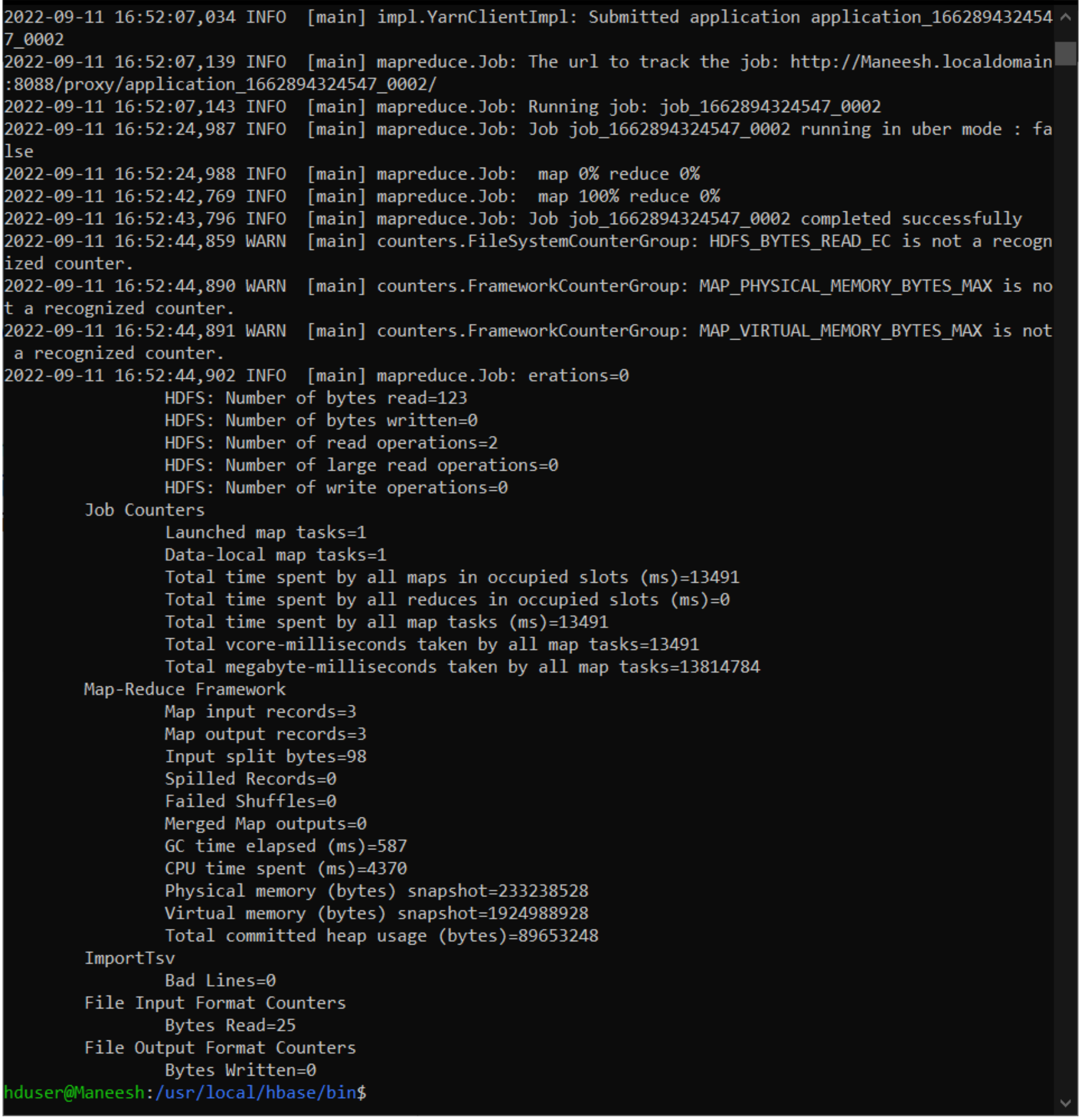


```
hduser@Maneesh:/usr/local$ hdfs dfs -copyFromLocal /usr/local/simple1.txt /
hduser@Maneesh:/usr/local$ cd /usr/local/hbase/bin
```

3. import simple1.txt to hbase.
   - ./hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator="," -Dimporttsv.columns=HBASE_ROW_KEY,cf test /simple1.txt



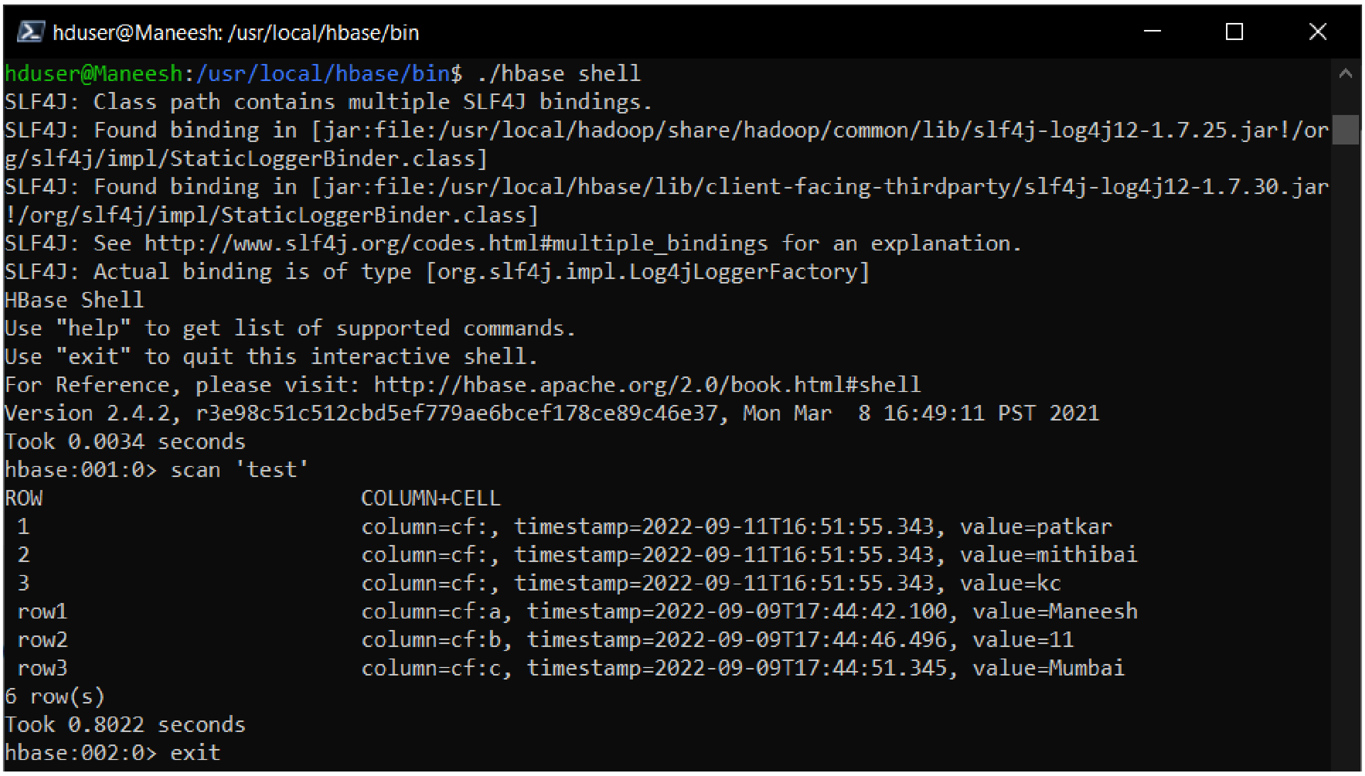
```
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.sep
arator="," -Dimporttsv.columns=HBASE_ROW_KEY,cf test /simple1.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/hadoop-3.3.0/share/hadoop/common/lib/slf4j-log4j12
-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2022-09-09 17:47:46,752 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: Client e
nvironment:zookeeper.version=3.5.7-f0fdd52973d373ffd9c86b81d99842dc2c7f660e, built on 02/10/2020 11:30 G
MT
2022-09-09 17:47:46,753 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: Client e
nvironment:host.name=Maneesh.localdomain
2022-09-09 17:47:46,753 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: Client e
nvironment:java.version=1.8.0_342
2022-09-09 17:47:46,753 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: Client e
nvironment:java.vendor=Private Build
2022-09-09 17:47:46,755 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: Client e
nvironment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre
2022-09-09 17:47:46,756 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x7c6908d7] zookeeper.ZooKeeper: hduser/h
adoop/hadoop-3.3.0/share/hadoop/yarn/hadoop-yarn-client-3.3.0.jar:/home/hduser/hadoop/hadoop-3.3.0/share
```



```
2022-09-11 16:52:07,034 INFO  [main] impl.YarnClientImpl: Submitted application application_166289432454
7_0002
2022-09-11 16:52:07,139 INFO  [main] mapreduce.Job: The url to track the job: http://Maneesh.localdomain
:8088/proxy/application_1662894324547_0002/
2022-09-11 16:52:07,143 INFO  [main] mapreduce.Job: Running job: job_1662894324547_0002
2022-09-11 16:52:24,987 INFO  [main] mapreduce.Job: Job job_1662894324547_0002 running in uber mode : fa
lse
2022-09-11 16:52:24,988 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2022-09-11 16:52:42,769 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2022-09-11 16:52:43,796 INFO  [main] mapreduce.Job: Job job_1662894324547_0002 completed successfully
2022-09-11 16:52:44,859 WARN  [main] counters.FileSystemCounterGroup: HDFS_BYTES_READ_EC is not a recogn
ized counter.
2022-09-11 16:52:44,890 WARN  [main] counters.FrameworkCounterGroup: MAP_PHYSICAL_MEMORY_BYTES_MAX is no
t a recognized counter.
2022-09-11 16:52:44,891 WARN  [main] counters.FrameworkCounterGroup: MAP_VIRTUAL_MEMORY_BYTES_MAX is not
 a recognized counter.
2022-09-11 16:52:44,902 INFO  [main] mapreduce.Job: erations=0
                HDFS: Number of bytes read=123
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=2
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=13491
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=13491
                Total vcore-milliseconds taken by all map tasks=13491
                Total megabyte-milliseconds taken by all map tasks=13814784
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Input split bytes=98
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=587
                CPU time spent (ms)=4370
                Physical memory (bytes) snapshot=233238528
                Virtual memory (bytes) snapshot=1924988928
                Total committed heap usage (bytes)=89653248
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=25
        File Output Format Counters
                Bytes Written=0
hduser@Maneesh:/usr/local/hbase/bin$
```

4. Start HBase Shell and check table contents
   - ➢ ./hbase shell

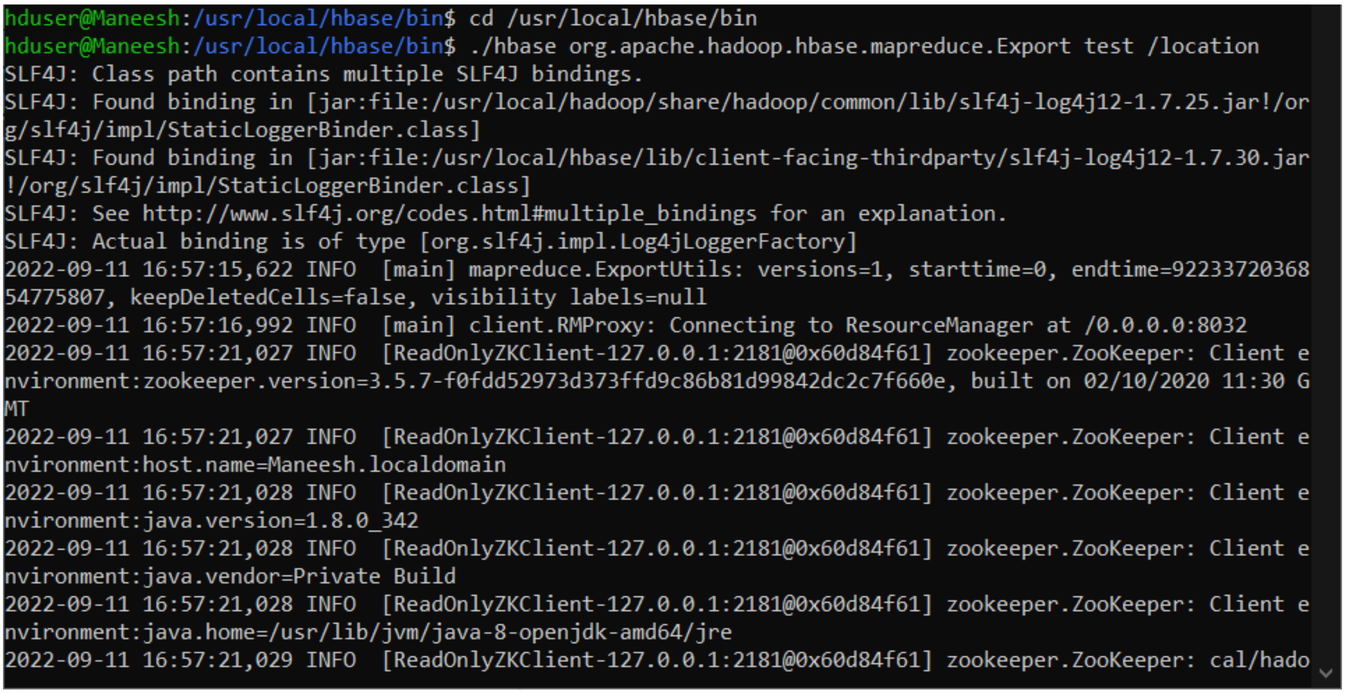
     scan 'test'
     exit



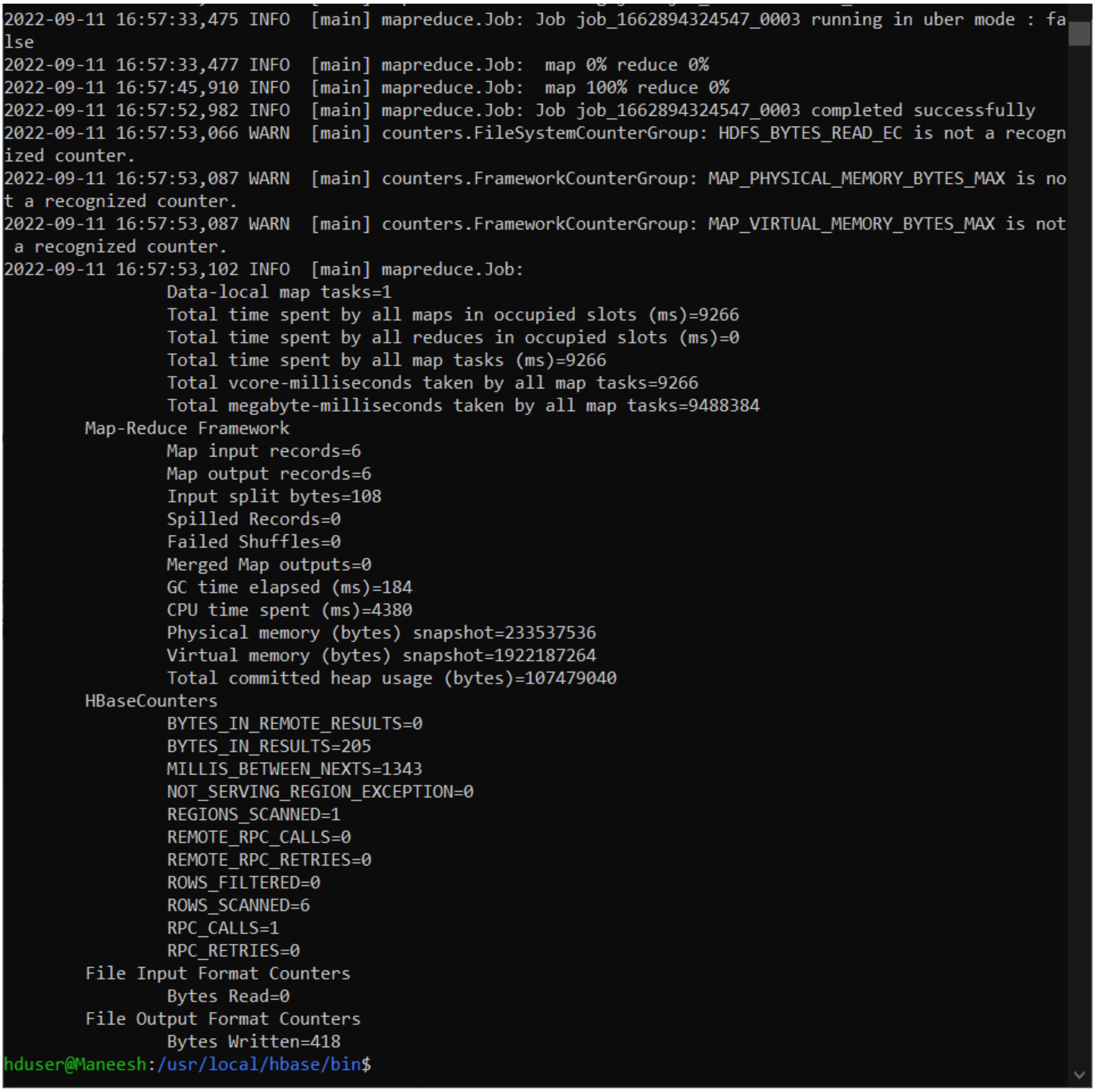
```
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.2, r3e98c51c512cbd5ef779ae6bcef178ce89c46e37, Mon Mar  8 16:49:11 PST 2021
Took 0.0034 seconds
hbase:001:0> scan 'test'
ROW                        COLUMN+CELL
 1                         column=cf:, timestamp=2022-09-11T16:51:55.343, value=patkar
 2                         column=cf:, timestamp=2022-09-11T16:51:55.343, value=mithibai
 3                         column=cf:, timestamp=2022-09-11T16:51:55.343, value=kc
 row1                      column=cf:a, timestamp=2022-09-09T17:44:42.100, value=Maneesh
 row2                      column=cf:b, timestamp=2022-09-09T17:44:46.496, value=11
 row3                      column=cf:c, timestamp=2022-09-09T17:44:51.345, value=Mumbai
6 row(s)
Took 0.8022 seconds
hbase:002:0> exit
```

TO EXPORT HBASE TABLE TO HDFS

- ➢ cd /usr/local/hbase/bin
- ➢ ./hbase org.apache.hadoop.hbase.mapreduce.Export test /location

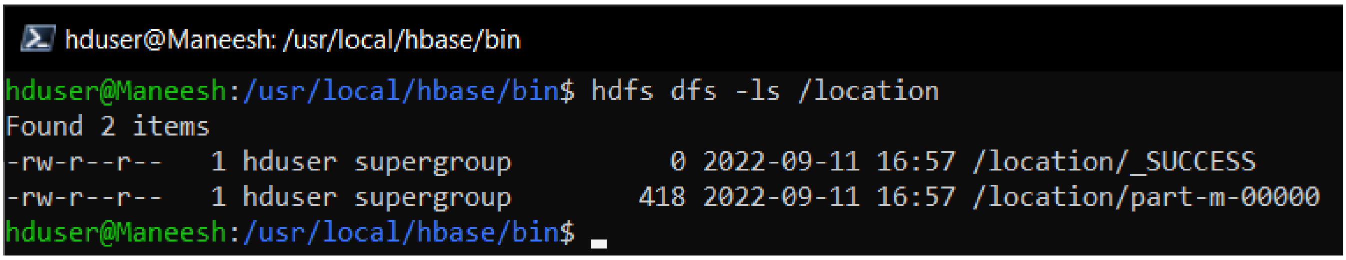


```
hduser@Maneesh:/usr/local/hbase/bin$ cd /usr/local/hbase/bin
hduser@Maneesh:/usr/local/hbase/bin$ ./hbase org.apache.hadoop.hbase.mapreduce.Export test /location
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2022-09-11 16:57:15,622 INFO  [main] mapreduce.ExportUtils: versions=1, starttime=0, endtime=92233720368
54775807, keepDeletedCells=false, visibility labels=null
2022-09-11 16:57:16,992 INFO  [main] client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2022-09-11 16:57:21,027 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: Client e
nvironment:zookeeper.version=3.5.7-f0fdd52973d373ffd9c86b81d99842dc2c7f660e, built on 02/10/2020 11:30 G
MT
2022-09-11 16:57:21,027 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: Client e
nvironment:host.name=Maneesh.localdomain
2022-09-11 16:57:21,028 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: Client e
nvironment:java.version=1.8.0_342
2022-09-11 16:57:21,028 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: Client e
nvironment:java.vendor=Private Build
2022-09-11 16:57:21,028 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: Client e
nvironment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre
2022-09-11 16:57:21,029 INFO  [ReadOnlyZKClient-127.0.0.1:2181@0x60d84f61] zookeeper.ZooKeeper: cal/hado
```



```
2022-09-11 16:57:33,475 INFO  [main] mapreduce.Job: Job job_1662894324547_0003 running in uber mode : fa
lse
2022-09-11 16:57:33,477 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2022-09-11 16:57:45,910 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2022-09-11 16:57:52,982 INFO  [main] mapreduce.Job: Job job_1662894324547_0003 completed successfully
2022-09-11 16:57:53,066 WARN  [main] counters.FileSystemCounterGroup: HDFS_BYTES_READ_EC is not a recogn
ized counter.
2022-09-11 16:57:53,087 WARN  [main] counters.FrameworkCounterGroup: MAP_PHYSICAL_MEMORY_BYTES_MAX is no
t a recognized counter.
2022-09-11 16:57:53,087 WARN  [main] counters.FrameworkCounterGroup: MAP_VIRTUAL_MEMORY_BYTES_MAX is not
 a recognized counter.
2022-09-11 16:57:53,102 INFO  [main] mapreduce.Job:
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=9266
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=9266
                Total vcore-milliseconds taken by all map tasks=9266
                Total megabyte-milliseconds taken by all map tasks=9488384
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Input split bytes=108
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=184
                CPU time spent (ms)=4380
                Physical memory (bytes) snapshot=233537536
                Virtual memory (bytes) snapshot=1922187264
                Total committed heap usage (bytes)=107479040
        HBaseCounters
                BYTES_IN_REMOTE_RESULTS=0
                BYTES_IN_RESULTS=205
                MILLIS_BETWEEN_NEXTS=1343
                NOT_SERVING_REGION_EXCEPTION=0
                REGIONS_SCANNED=1
                REMOTE_RPC_CALLS=0
                REMOTE_RPC_RETRIES=0
                ROWS_FILTERED=0
                ROWS_SCANNED=6
                RPC_CALLS=1
                RPC_RETRIES=0
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=418
hduser@Maneesh:/usr/local/hbase/bin$
```

- ➢ hdfs dfs -ls /location



```
hduser@Maneesh:/usr/local/hbase/bin$ hdfs dfs -ls /location
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-09-11 16:57 /location/_SUCCESS
-rw-r--r--   1 hduser supergroup        418 2022-09-11 16:57 /location/part-m-00000
hduser@Maneesh:/usr/local/hbase/bin$ _
```

Practical No: 6

Aim: Install and configure PIG.

Theory:

Pig Hadoop is basically a high-level programming language that is helpful for the analysis of huge datasets. Pig Hadoop was developed by Yahoo! and is generally used with Hadoop to perform a lot of data administration operations.

For writing data analysis programs, Pig renders a high-level programming language called Pig Latin. Several operators are provided by Pig Latin using which personalized functions for writing, reading, and processing of data can be developed by programmers.

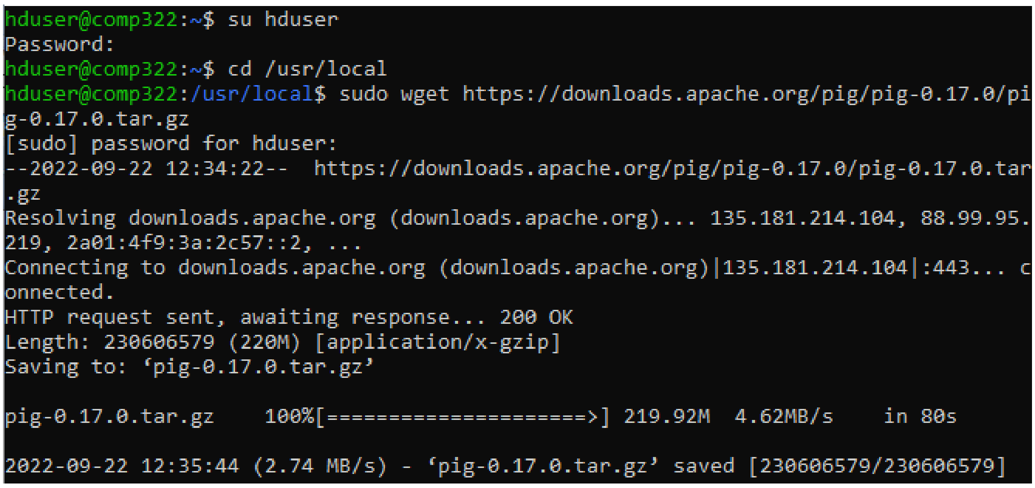
For analyzing data through Apache Pig, we need to write scripts using Pig Latin. Then, these scripts need to be transformed into MapReduce tasks. This is achieved with the help of Pig Engine.

Features of Pig Hadoop:

- In-built operators: Apache Pig provides a very good set of operators for performing several data operations like sort, join, filter, etc.
- Ease of programming: Since Pig Latin has similarities with SQL, it is very easy to write a Pig script.
- Automatic optimization: The tasks in Apache Pig are automatically optimized. This makes the programmers concentrate only on the semantics of the language.
- Handles all kinds of data: Apache Pig can analyze both structured and unstructured data and store the results in HDFS.

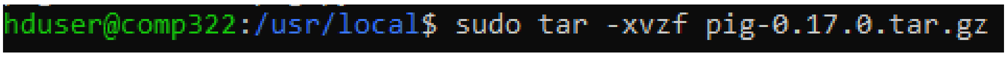
Steps:

1. Installation Part:
   - su hduser
   - cd /usr/local
   - sudo wget https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz



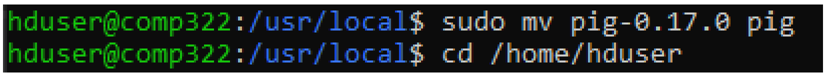
```
hduser@comp322:~$ su hduser
Password:
hduser@comp322:~$ cd /usr/local
hduser@comp322:/usr/local$ sudo wget https://downloads.apache.org/pig/pig-0.17.0/pi
g-0.17.0.tar.gz
[sudo] password for hduser:
--2022-09-22 12:34:22--  https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar
.gz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.95.
219, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 230606579 (220M) [application/x-gzip]
Saving to: ‘pig-0.17.0.tar.gz’

pig-0.17.0.tar.gz    100%[===================>] 219.92M  4.62MB/s    in 80s

2022-09-22 12:35:44 (2.74 MB/s) - ‘pig-0.17.0.tar.gz’ saved [230606579/230606579]
```

   - sudo tar -xvzf pig-0.17.0.tar.gz



```
hduser@comp322:/usr/local$ sudo tar -xvzf pig-0.17.0.tar.gz
```

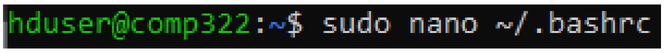
   - sudo mv pig-0.17.0 pig
   - cd /home/hduser



```
hduser@comp322:/usr/local$ sudo mv pig-0.17.0 pig
hduser@comp322:/usr/local$ cd /home/hduser
```

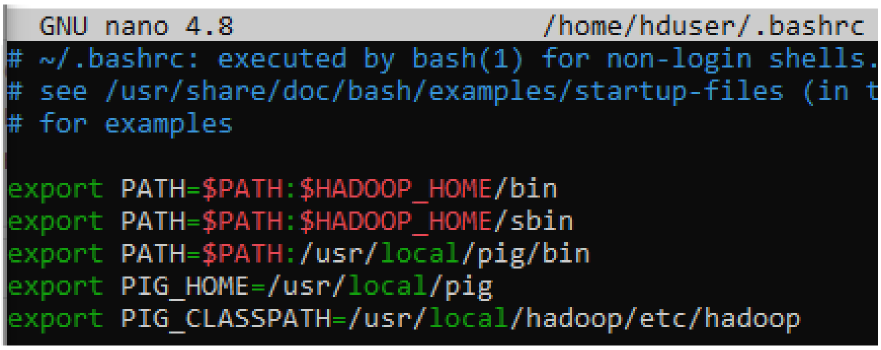
2. In the sudo nano ~/.bashrc file, add these lines very carefully:

- sudo nano ~/.bashrc



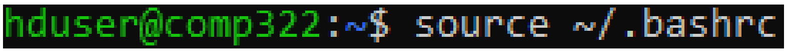
```
hduser@comp322:~$ sudo nano ~/.bashrc
```

export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export PATH=$PATH:/usr/local/pig/bin
export PIG_HOME=/usr/local/pig
export PIG_CLASSPATH=/usr/local/hadoop/etc/hadoop



```
  GNU nano 4.8                      /home/hduser/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in t
# for examples

export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export PATH=$PATH:/usr/local/pig/bin
export PIG_HOME=/usr/local/pig
export PIG_CLASSPATH=/usr/local/hadoop/etc/hadoop
```

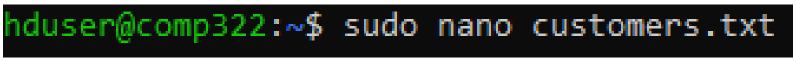
3. Then run command:
   - source ~/.bashrc



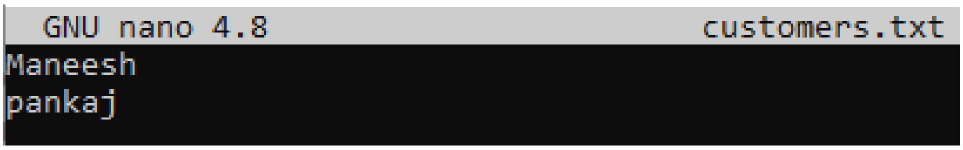
```
hduser@comp322:~$ source ~/.bashrc
```

Pig is installed

4. Create a file called sudo nano customers.txt and save it in any directory
   1) Running in local mode (pig can access data present only in local file system, eg the customer file)
   - sudo nano customers.txt



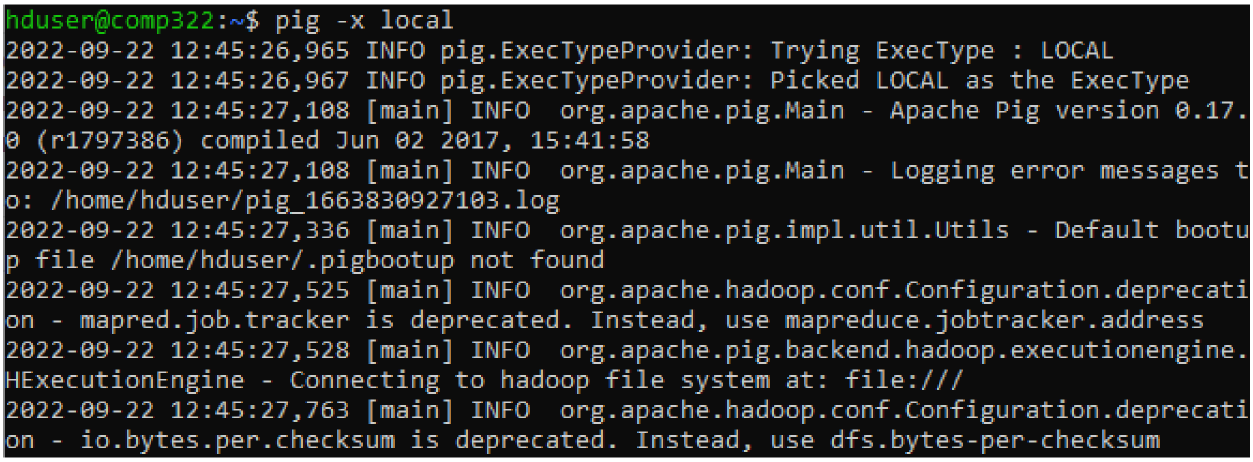
```
hduser@comp322:~$ sudo nano customers.txt
```



```
  GNU nano 4.8                           customers.txt
Maneesh
pankaj
```

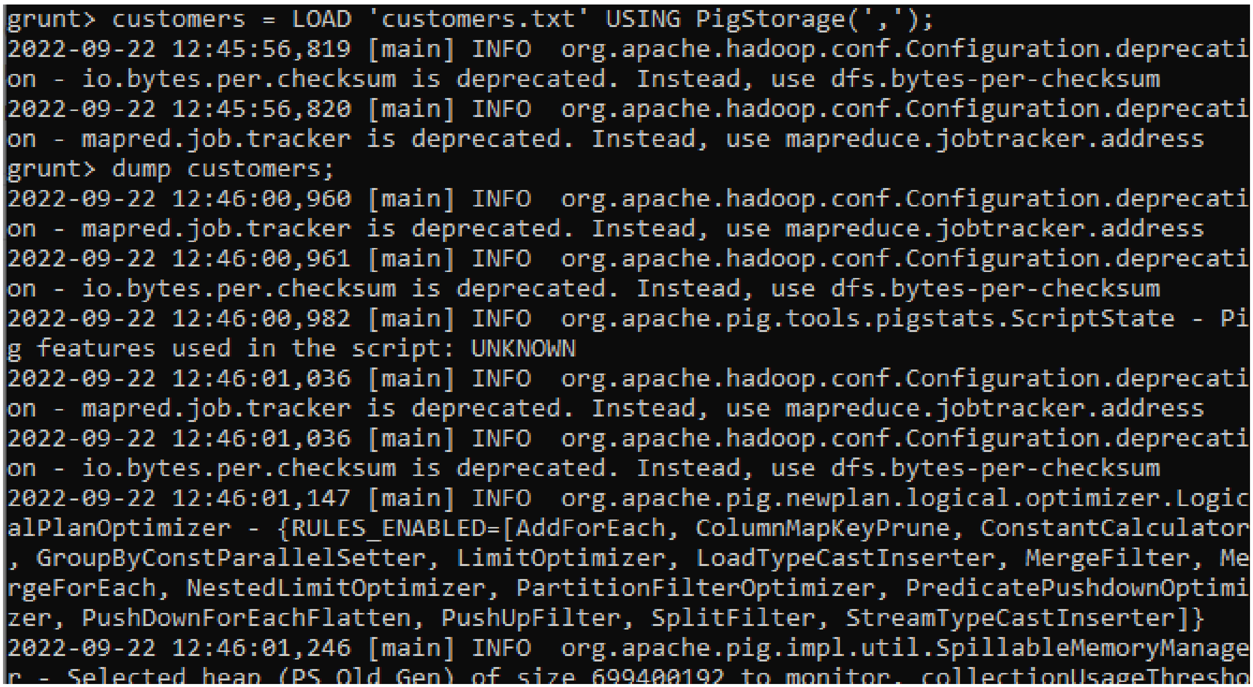
Start pig using following command:

- pig -x local

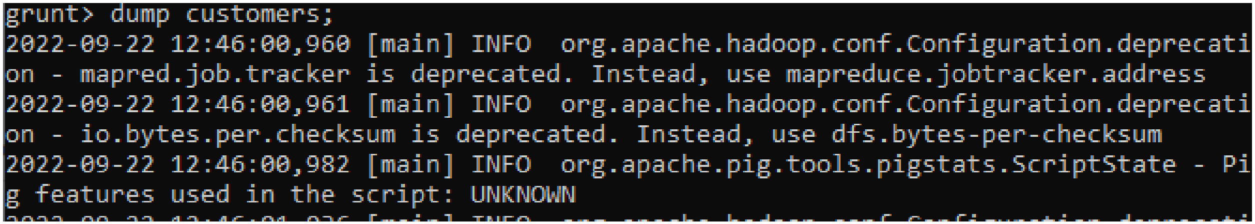


```
hduser@comp322:~$ pig -x local
2022-09-22 12:45:26,965 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2022-09-22 12:45:26,967 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2022-09-22 12:45:27,108 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.
0 (r1797386) compiled Jun 02 2017, 15:41:58
2022-09-22 12:45:27,108 [main] INFO  org.apache.pig.Main - Logging error messages t
o: /home/hduser/pig_1663830927103.log
2022-09-22 12:45:27,336 [main] INFO  org.apache.pig.impl.util.Utils - Default bootu
p file /home/hduser/.pigbootup not found
2022-09-22 12:45:27,525 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2022-09-22 12:45:27,528 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
HExecutionEngine - Connecting to hadoop file system at: file:///
2022-09-22 12:45:27,763 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
```

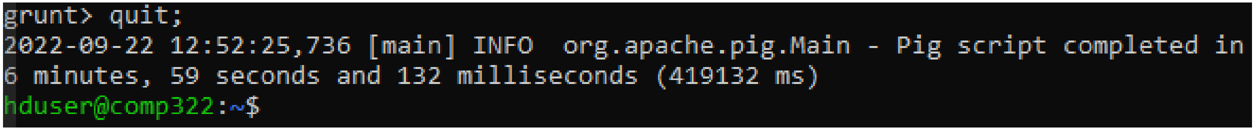
customers = LOAD 'customers.txt' USING PigStorage(',');
dump customers;
quit;



```
grunt> customers = LOAD 'customers.txt' USING PigStorage(',');
2022-09-22 12:45:56,819 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-09-22 12:45:56,820 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
grunt> dump customers;
2022-09-22 12:46:00,960 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2022-09-22 12:46:00,961 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-09-22 12:46:00,982 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pi
g features used in the script: UNKNOWN
2022-09-22 12:46:01,036 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2022-09-22 12:46:01,036 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-09-22 12:46:01,147 [main] INFO  org.apache.pig.newplan.logical.optimizer.Logic
alPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalculator
, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, Me
rgeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimi
zer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2022-09-22 12:46:01,246 [main] INFO  org.apache.pig.impl.util.SpillableMemoryManage
r - Selected heap (PS Old Gen) of size 699400192 to monitor. collectionUsageThresho
```



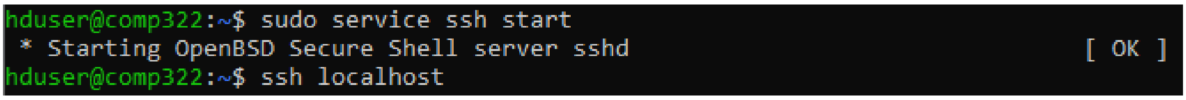
```
grunt> dump customers;
2022-09-22 12:46:00,960 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2022-09-22 12:46:00,961 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-09-22 12:46:00,982 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pi
g features used in the script: UNKNOWN
2022-09-22 12:46:01,036 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
```



```
grunt> quit;
2022-09-22 12:52:25,736 [main] INFO  org.apache.pig.Main - Pig script completed in
6 minutes, 59 seconds and 132 milliseconds (419132 ms)
hduser@comp322:~$
```

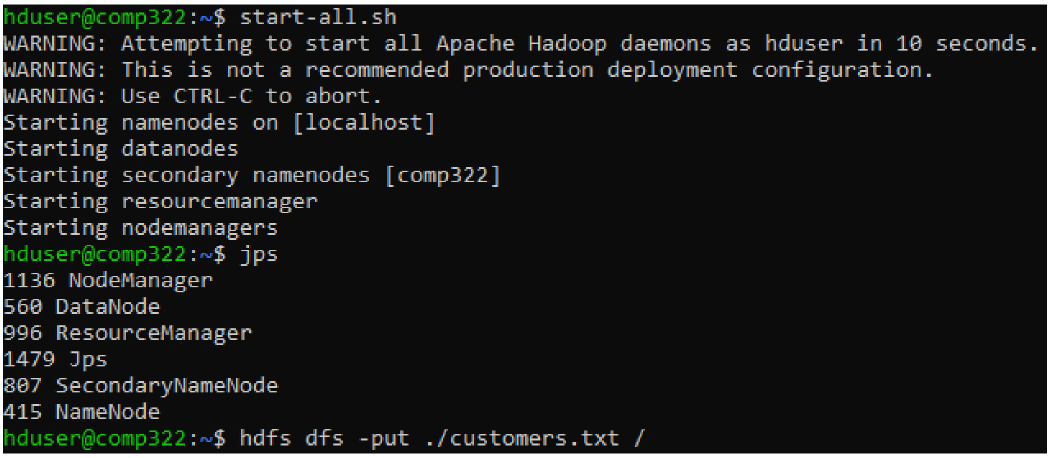
2) Running in HDFS mode (pig can access data on HDFS)
First we need to move customers.txt to HDFS
For that start hadoop,

- sudo service ssh start
- ssh localhost



```
hduser@comp322:~$ sudo service ssh start
 * Starting OpenBSD Secure Shell server sshd                                [ OK ]
hduser@comp322:~$ ssh localhost
```

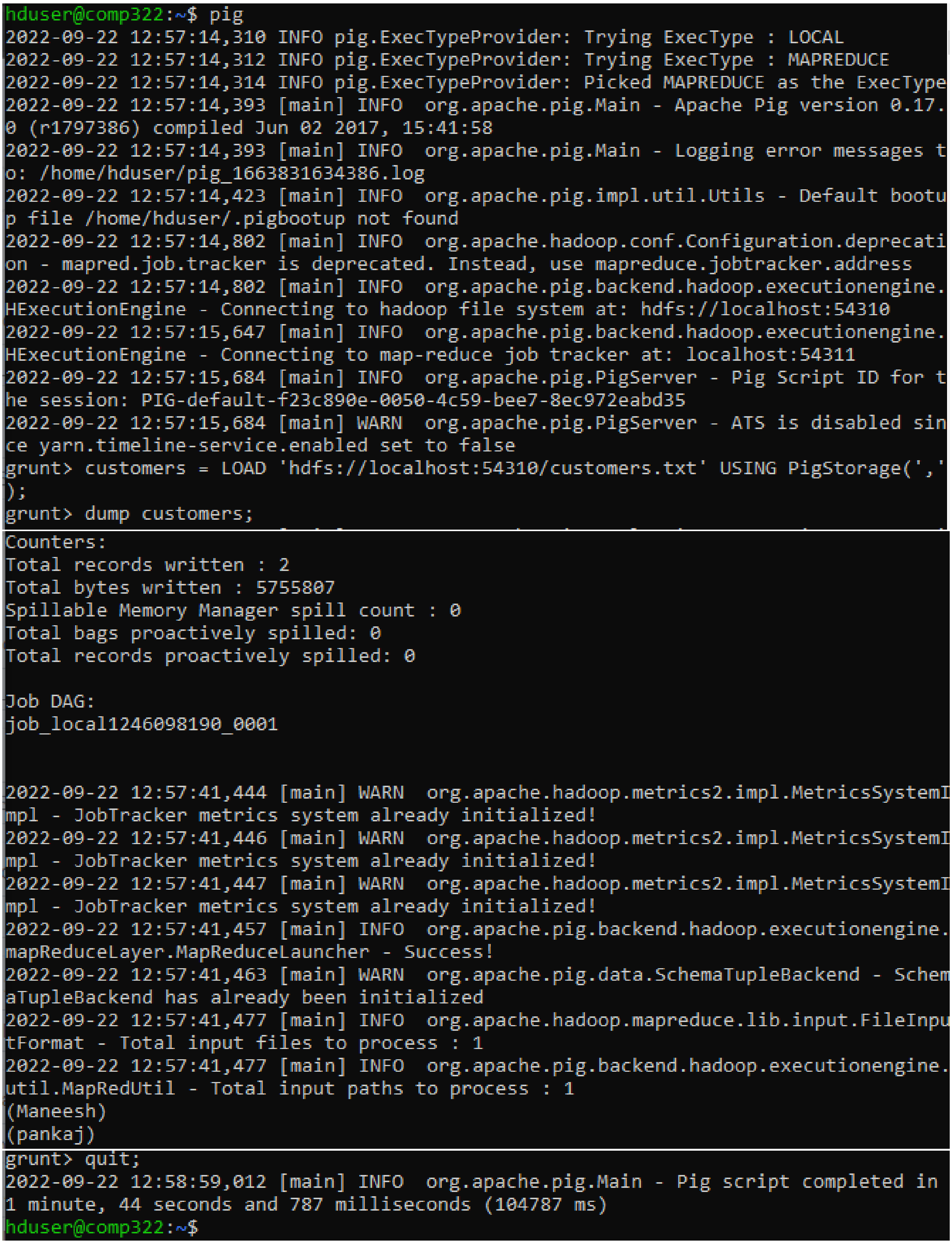
- start-all.sh
- jps
- hdfs dfs -put ./customers.txt /



```
hduser@comp322:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [comp322]
Starting resourcemanager
Starting nodemanagers
hduser@comp322:~$ jps
1136 NodeManager
560 DataNode
996 ResourceManager
1479 Jps
807 SecondaryNameNode
415 NameNode
hduser@comp322:~$ hdfs dfs -put ./customers.txt /
```

5. Start pig using pig command
   - ➢ pig

     customers = LOAD 'hdfs://localhost:54310/customers.txt' USING PigStorage(',');
     dump customers;
     quit;



```
hduser@comp322:~$ pig
2022-09-22 12:57:14,310 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2022-09-22 12:57:14,312 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2022-09-22 12:57:14,314 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2022-09-22 12:57:14,393 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.
0 (r1797386) compiled Jun 02 2017, 15:41:58
2022-09-22 12:57:14,393 [main] INFO  org.apache.pig.Main - Logging error messages t
o: /home/hduser/pig_1663831634386.log
2022-09-22 12:57:14,423 [main] INFO  org.apache.pig.impl.util.Utils - Default bootu
p file /home/hduser/.pigbootup not found
2022-09-22 12:57:14,802 [main] INFO  org.apache.hadoop.conf.Configuration.deprecati
on - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2022-09-22 12:57:14,802 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:54310
2022-09-22 12:57:15,647 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
HExecutionEngine - Connecting to map-reduce job tracker at: localhost:54311
2022-09-22 12:57:15,684 [main] INFO  org.apache.pig.PigServer - Pig Script ID for t
he session: PIG-default-f23c890e-0050-4c59-bee7-8ec972eabd35
2022-09-22 12:57:15,684 [main] WARN  org.apache.pig.PigServer - ATS is disabled sin
ce yarn.timeline-service.enabled set to false
grunt> customers = LOAD 'hdfs://localhost:54310/customers.txt' USING PigStorage(','
);
grunt> dump customers;
```

```
Counters:
Total records written : 2
Total bytes written : 5755807
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1246098190_0001


2022-09-22 12:57:41,444 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemI
mpl - JobTracker metrics system already initialized!
2022-09-22 12:57:41,446 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemI
mpl - JobTracker metrics system already initialized!
2022-09-22 12:57:41,447 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemI
mpl - JobTracker metrics system already initialized!
2022-09-22 12:57:41,457 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
mapReduceLayer.MapReduceLauncher - Success!
2022-09-22 12:57:41,463 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Schem
aTupleBackend has already been initialized
2022-09-22 12:57:41,477 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInpu
tFormat - Total input files to process : 1
2022-09-22 12:57:41,477 [main] INFO  org.apache.pig.backend.hadoop.executionengine.
util.MapRedUtil - Total input paths to process : 1
(Maneesh)
(pankaj)
```

```
grunt> quit;
2022-09-22 12:58:59,012 [main] INFO  org.apache.pig.Main - Pig script completed in
1 minute, 44 seconds and 787 milliseconds (104787 ms)
hduser@comp322:~$
```

Practical No: 7

Aim: Perform importing and exporting of data between SQL and Hadoop using Sqoop.

Theory:

Apache SQOOP (SQL-to-Hadoop) is a tool designed to support bulk export and import of data into HDFS from structured data stores such as relational databases, enterprise data warehouses, and NoSQL systems. It is a data migration tool based upon a connector architecture which supports plugins to provide connectivity to new external systems.

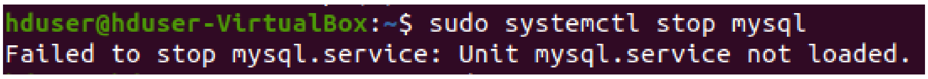
An example use case of Hadoop Sqoop is an enterprise that runs a nightly Sqoop import to load the day’s data from a production transactional RDBMS into a Hive data warehouse for further analysis.

Some of the important Features of the Sqoop:

- Sqoop also helps us to connect the result from the SQL Queries into Hadoop distributed file system.
- Sqoop helps us to load the processed data directly into the hive or Hbase.
- It performs the security operation of data with the help of Kerberos.
- With the help of Sqoop, we can perform compression of processed data.
- Sqoop is highly powerful and efficient in nature.

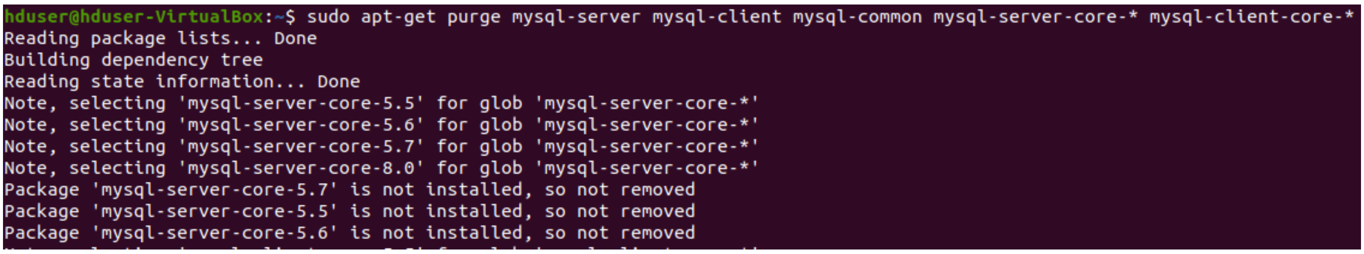
Steps:

1. To initially remove MySQL:
   - sudo systemctl stop mysql



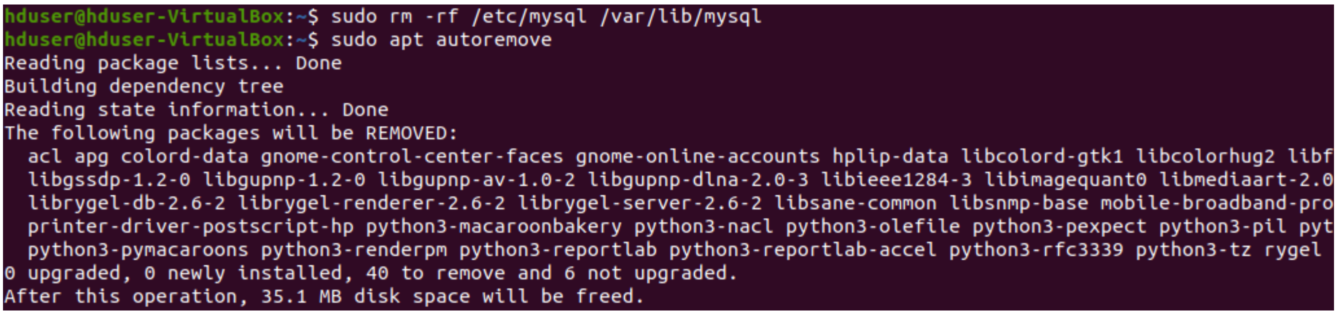
```
hduser@hduser-VirtualBox:~$ sudo systemctl stop mysql
Failed to stop mysql.service: Unit mysql.service not loaded.
```

- sudo apt-get purge mysql-server mysql-client mysql-common mysql-server-core-* mysql-client-core-*



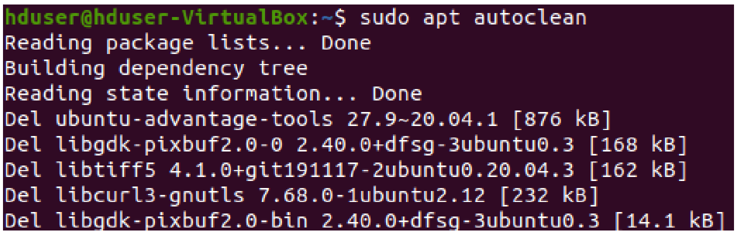
```
hduser@hduser-VirtualBox:~$ sudo apt-get purge mysql-server mysql-client mysql-common mysql-server-core-* mysql-client-core-*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'mysql-server-core-5.5' for glob 'mysql-server-core-*'
Note, selecting 'mysql-server-core-5.6' for glob 'mysql-server-core-*'
Note, selecting 'mysql-server-core-5.7' for glob 'mysql-server-core-*'
Note, selecting 'mysql-server-core-8.0' for glob 'mysql-server-core-*'
Package 'mysql-server-core-5.7' is not installed, so not removed
Package 'mysql-server-core-5.5' is not installed, so not removed
Package 'mysql-server-core-5.6' is not installed, so not removed
```

- sudo rm -rf /etc/mysql /var/lib/mysql
- sudo apt autoremove



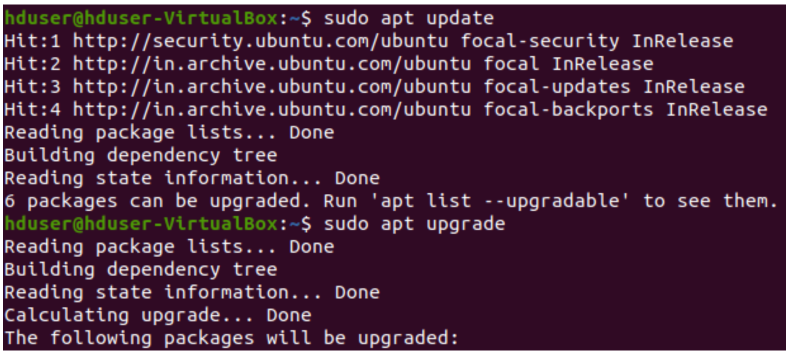
```
hduser@hduser-VirtualBox:~$ sudo rm -rf /etc/mysql /var/lib/mysql
hduser@hduser-VirtualBox:~$ sudo apt autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  acl apg colord-data gnome-control-center-faces gnome-online-accounts hplip-data libcolord-gtk1 libcolorhug2 libf
  libgssdp-1.2-0 libgupnp-1.2-0 libgupnp-av-1.0-2 libgupnp-dlna-2.0-3 libieee1284-3 libimagequant0 libmediaart-2.0
  librygel-db-2.6-2 librygel-renderer-2.6-2 librygel-server-2.6-2 libsane-common libsnmp-base mobile-broadband-pro
  printer-driver-postscript-hp python3-macaroonbakery python3-nacl python3-olefile python3-pexpect python3-pil pyt
  python3-pymacaroons python3-renderpm python3-reportlab python3-reportlab-accel python3-rfc3339 python3-tz rygel
0 upgraded, 0 newly installed, 40 to remove and 6 not upgraded.
After this operation, 35.1 MB disk space will be freed.
```

- sudo apt autoclean



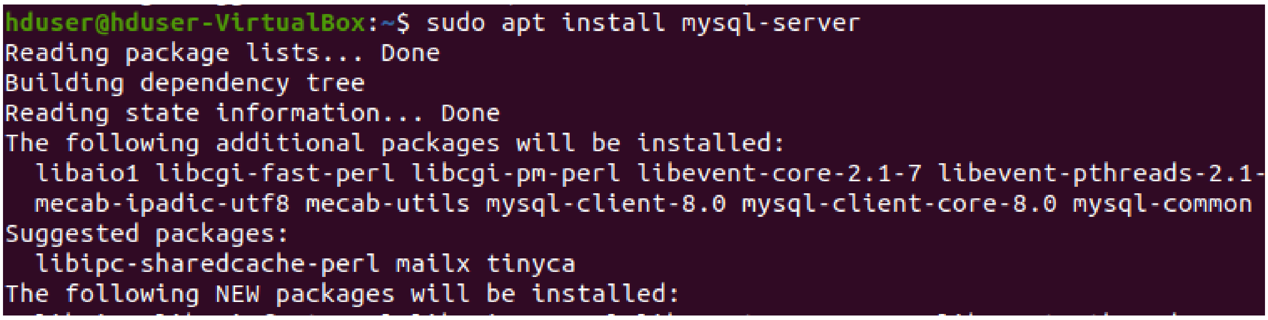
```
hduser@hduser-VirtualBox:~$ sudo apt autoclean
Reading package lists... Done
Building dependency tree
Reading state information... Done
Del ubuntu-advantage-tools 27.9~20.04.1 [876 kB]
Del libgdk-pixbuf2.0-0 2.40.0+dfsg-3ubuntu0.3 [168 kB]
Del libtiff5 4.1.0+git191117-2ubuntu0.20.04.3 [162 kB]
Del libcurl3-gnutls 7.68.0-1ubuntu2.12 [232 kB]
Del libgdk-pixbuf2.0-bin 2.40.0+dfsg-3ubuntu0.3 [14.1 kB]
```

2. we install MySQL:
   - ➢ sudo apt update
   - ➢ sudo apt upgrade



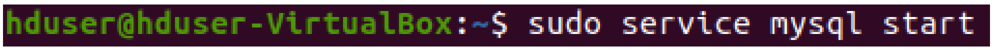
```
hduser@hduser-VirtualBox:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
hduser@hduser-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
```

   - ➢ sudo apt install mysql-server



```
hduser@hduser-VirtualBox:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7 libevent-pthreads-2.1-
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
```

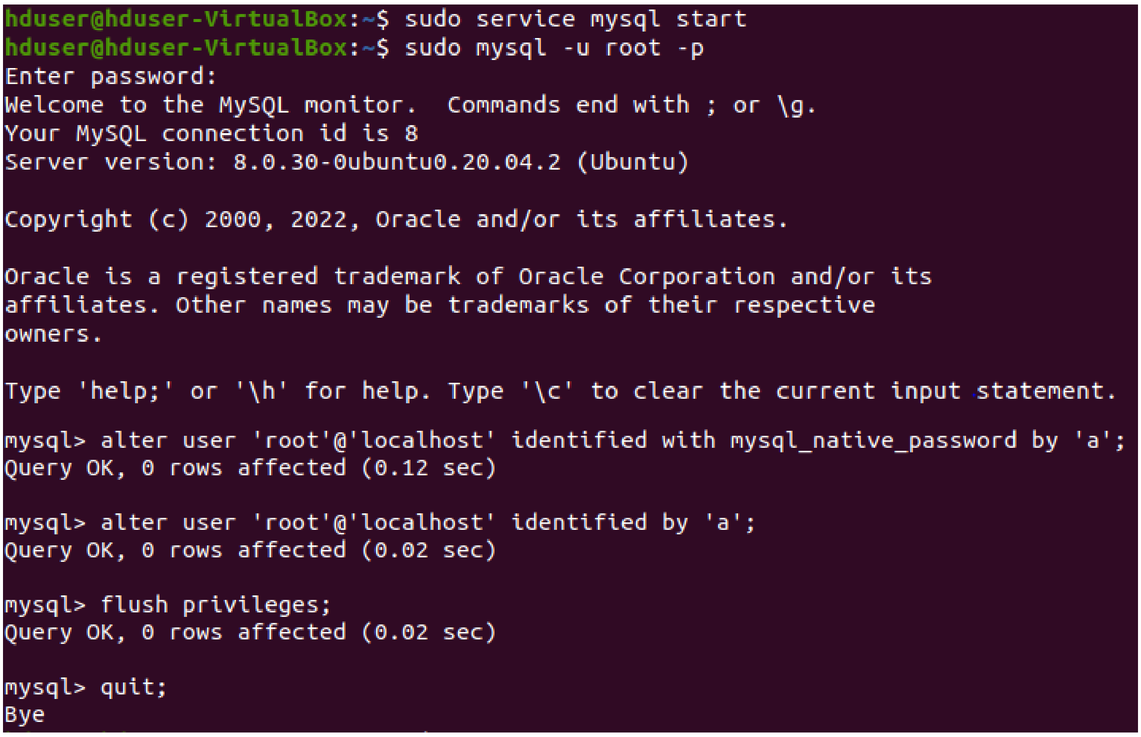
   - ➢ sudo service mysql start



```
hduser@hduser-VirtualBox:~$ sudo service mysql start
```

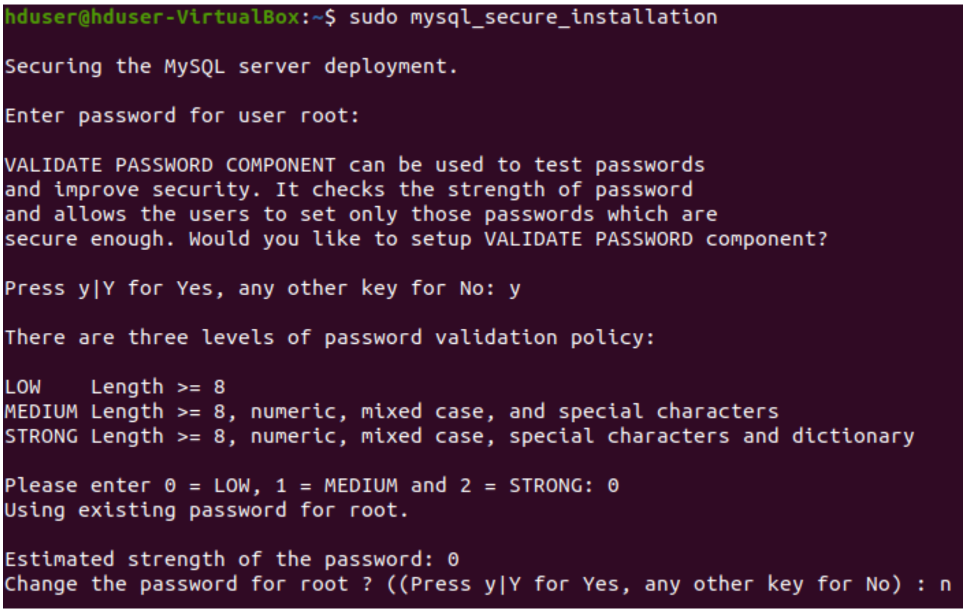
3. We set a few permissions:
   - ➢ sudo mysql -u root -p

     alter user 'root'@'localhost' identified with mysql_native_password by 'a';
     alter user 'root'@'localhost' identified by 'a';
     flush privileges;
     quit;

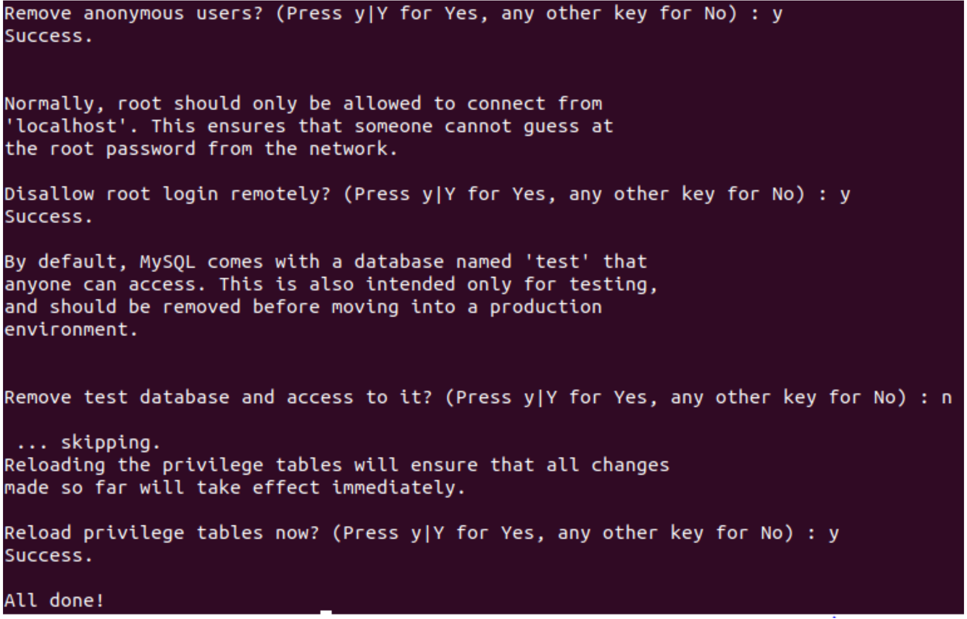


```
hduser@hduser-VirtualBox:~$ sudo service mysql start
hduser@hduser-VirtualBox:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> alter user 'root'@'localhost' identified with mysql_native_password by 'a';
Query OK, 0 rows affected (0.12 sec)

mysql> alter user 'root'@'localhost' identified by 'a';
Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

mysql> quit;
Bye
```

- ➢ sudo mysql_secure_installation
  - ▪ Enter password for user root: a
  - ▪ setup Vlidate password component Press y|Y for Yes, any other key for No: y
  - ▪ Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
  - ▪ Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
  - ▪ Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
  - ▪ Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
  - ▪ Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
  - ▪ Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y

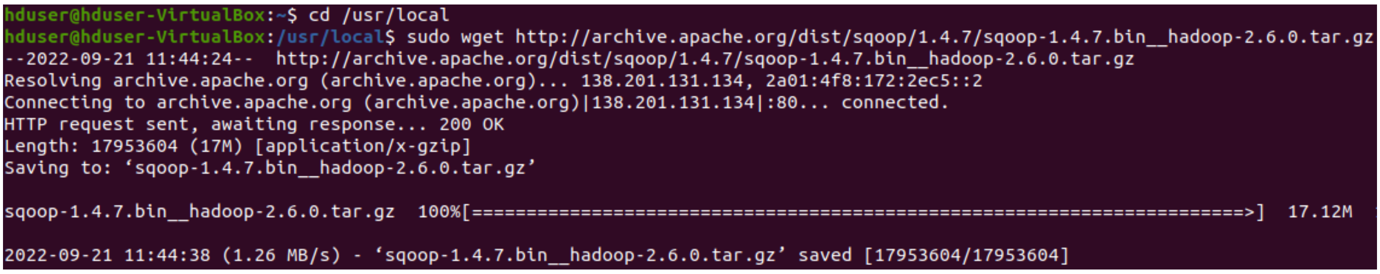


```
hduser@hduser-VirtualBox:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Using existing password for root.

Estimated strength of the password: 0
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```



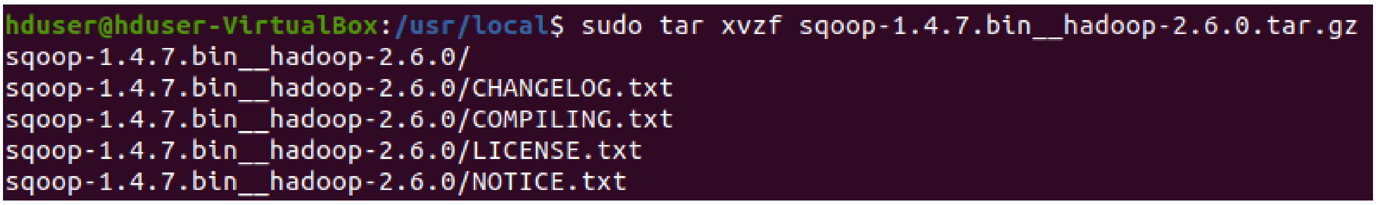
```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n

 ... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

4. Now we download and install sqoop:
   - ➢ cd /usr/local
   - ➢ sudo wget http://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz



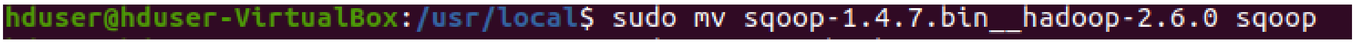
```
hduser@hduser-VirtualBox:~$ cd /usr/local
hduser@hduser-VirtualBox:/usr/local$ sudo wget http://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
--2022-09-21 11:44:24--  http://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17953604 (17M) [application/x-gzip]
Saving to: ‘sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz’

sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz  100%[=======================================================================>]  17.12M

2022-09-21 11:44:38 (1.26 MB/s) - ‘sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz’ saved [17953604/17953604]
```

- ➢ sudo tar xvzf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz



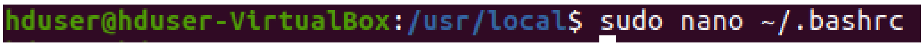
```
hduser@hduser-VirtualBox:/usr/local$ sudo tar xvzf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz
sqoop-1.4.7.bin__hadoop-2.6.0/
sqoop-1.4.7.bin__hadoop-2.6.0/CHANGELOG.txt
sqoop-1.4.7.bin__hadoop-2.6.0/COMPILING.txt
sqoop-1.4.7.bin__hadoop-2.6.0/LICENSE.txt
sqoop-1.4.7.bin__hadoop-2.6.0/NOTICE.txt
```

- ➢ sudo mv sqoop-1.4.7.bin__hadoop-2.6.0 sqoop



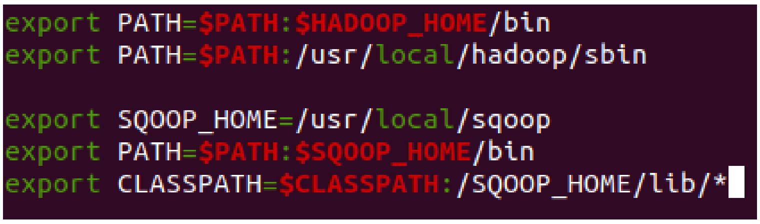
```
hduser@hduser-VirtualBox:/usr/local$ sudo mv sqoop-1.4.7.bin__hadoop-2.6.0 sqoop
```

5. In the sudo nano ~/.bashrc make the following changes:
   - ➢ sudo nano ~/.bashrc



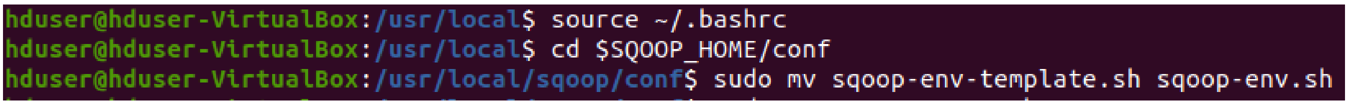
```
hduser@hduser-VirtualBox:/usr/local$ sudo nano ~/.bashrc
```

export SQOOP_HOME=/usr/local/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export CLASSPATH=$CLASSPATH:/SQOOP_HOME/lib/*



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

export SQOOP_HOME=/usr/local/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export CLASSPATH=$CLASSPATH:/SQOOP_HOME/lib/*
```

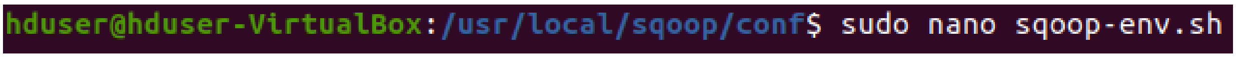
- ➢ source ~/.bashrc
- ➢ cd $SQOOP_HOME/conf
- ➢ sudo mv sqoop-env-template.sh sqoop-env.sh



```
hduser@hduser-VirtualBox:/usr/local$ source ~/.bashrc
hduser@hduser-VirtualBox:/usr/local$ cd $SQOOP_HOME/conf
hduser@hduser-VirtualBox:/usr/local/sqoop/conf$ sudo mv sqoop-env-template.sh sqoop-env.sh
```

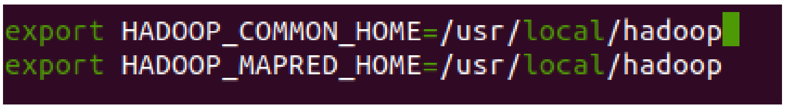
Add the following lines in sqoop-env.sh:

- ➢ sudo nano sqoop-env.sh



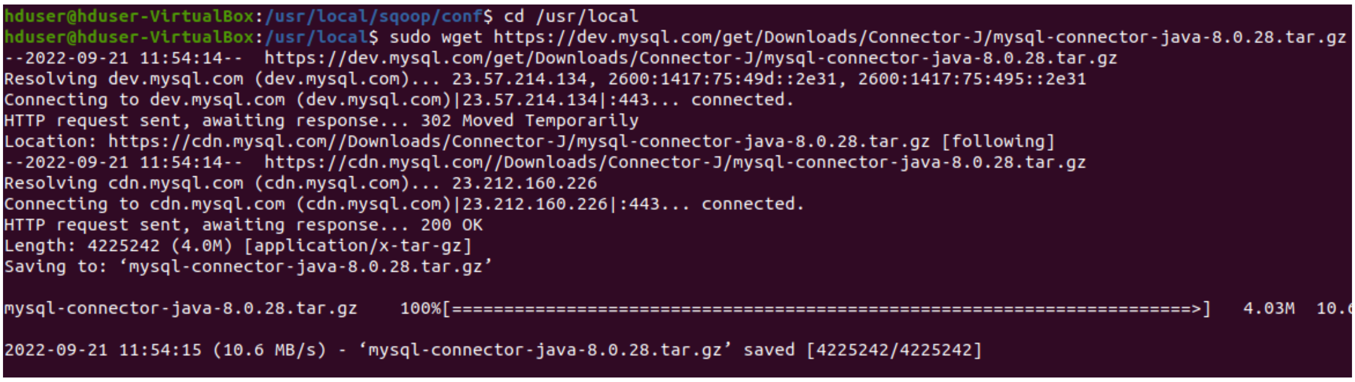
```
hduser@hduser-VirtualBox:/usr/local/sqoop/conf$ sudo nano sqoop-env.sh
```

export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop



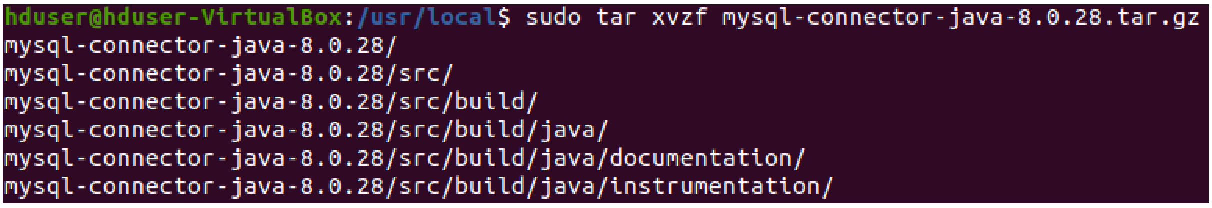
```
export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop
```

6. Now we need to download the appropriate MySQL java connector:
   - ➢ cd /usr/local
   - ➢ sudo wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.28.tar.gz



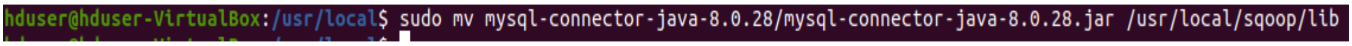
```
hduser@hduser-VirtualBox:/usr/local/sqoop/conf$ cd /usr/local
hduser@hduser-VirtualBox:/usr/local$ sudo wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.28.tar.gz
--2022-09-21 11:54:14--  https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.28.tar.gz
Resolving dev.mysql.com (dev.mysql.com)... 23.57.214.134, 2600:1417:75:49d::2e31, 2600:1417:75:495::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.57.214.134|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-java-8.0.28.tar.gz [following]
--2022-09-21 11:54:14--  https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-java-8.0.28.tar.gz
Resolving cdn.mysql.com (cdn.mysql.com)... 23.212.160.226
Connecting to cdn.mysql.com (cdn.mysql.com)|23.212.160.226|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4225242 (4.0M) [application/x-tar-gz]
Saving to: ‘mysql-connector-java-8.0.28.tar.gz’

mysql-connector-java-8.0.28.tar.gz    100%[==================================================================>]   4.03M  10.

2022-09-21 11:54:15 (10.6 MB/s) - ‘mysql-connector-java-8.0.28.tar.gz’ saved [4225242/4225242]
```

- sudo tar xvzf mysql-connector-java-8.0.28.tar.gz



```
hduser@hduser-VirtualBox:/usr/local$ sudo tar xvzf mysql-connector-java-8.0.28.tar.gz
mysql-connector-java-8.0.28/
mysql-connector-java-8.0.28/src/
mysql-connector-java-8.0.28/src/build/
mysql-connector-java-8.0.28/src/build/java/
mysql-connector-java-8.0.28/src/build/java/documentation/
mysql-connector-java-8.0.28/src/build/java/instrumentation/
```

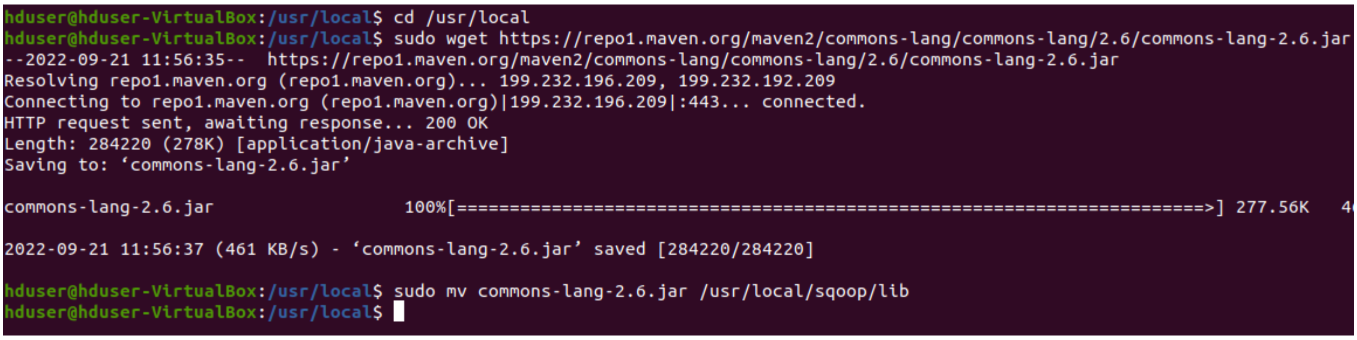
- sudo mv mysql-connector-java-8.0.28/mysql-connector-java-8.0.28.jar /usr/local/sqoop/lib



```
hduser@hduser-VirtualBox:/usr/local$ sudo mv mysql-connector-java-8.0.28/mysql-connector-java-8.0.28.jar /usr/local/sqoop/lib
```

We also need to download additional dependency

- cd /usr/local
- sudo wget https://repo1.maven.org/maven2/commons-lang/commons-lang/2.6/commons-lang-2.6.jar
- sudo mv commons-lang-2.6.jar /usr/local/sqoop/lib



```
hduser@hduser-VirtualBox:/usr/local$ cd /usr/local
hduser@hduser-VirtualBox:/usr/local$ sudo wget https://repo1.maven.org/maven2/commons-lang/commons-lang/2.6/commons-lang-2.6.jar
--2022-09-21 11:56:35--  https://repo1.maven.org/maven2/commons-lang/commons-lang/2.6/commons-lang-2.6.jar
Resolving repo1.maven.org (repo1.maven.org)... 199.232.196.209, 199.232.192.209
Connecting to repo1.maven.org (repo1.maven.org)|199.232.196.209|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 284220 (278K) [application/java-archive]
Saving to: ‘commons-lang-2.6.jar’

commons-lang-2.6.jar                100%[===================================================================>] 277.56K   4

2022-09-21 11:56:37 (461 KB/s) - ‘commons-lang-2.6.jar’ saved [284220/284220]

hduser@hduser-VirtualBox:/usr/local$ sudo mv commons-lang-2.6.jar /usr/local/sqoop/lib
hduser@hduser-VirtualBox:/usr/local$ 
```

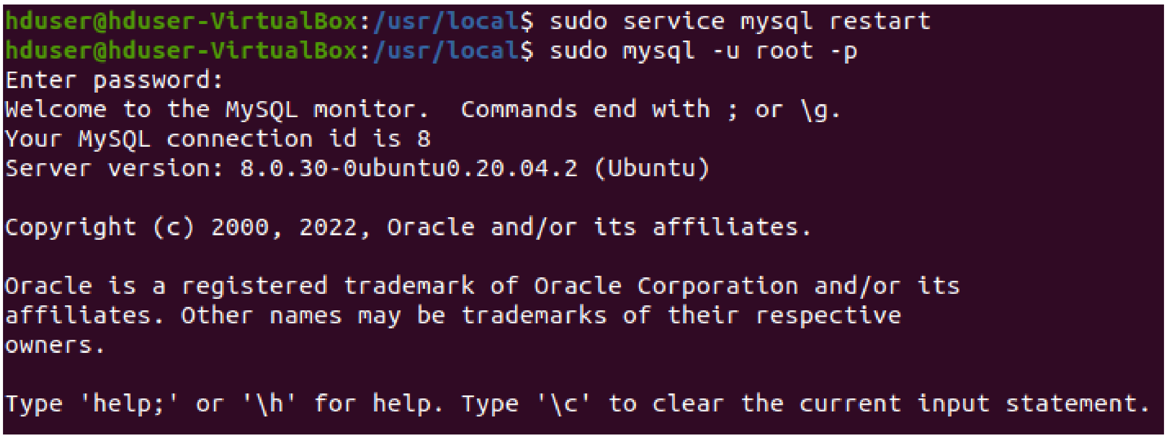
Sqoop installation completes here

----

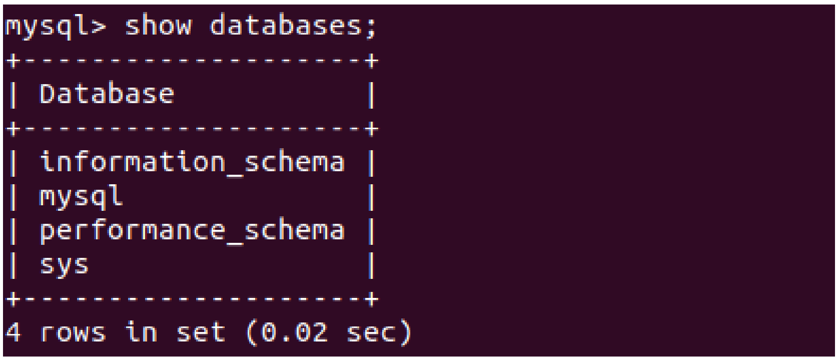
1. Create a table in MySQL:

- sudo service mysql restart
- sudo mysql -u root -p

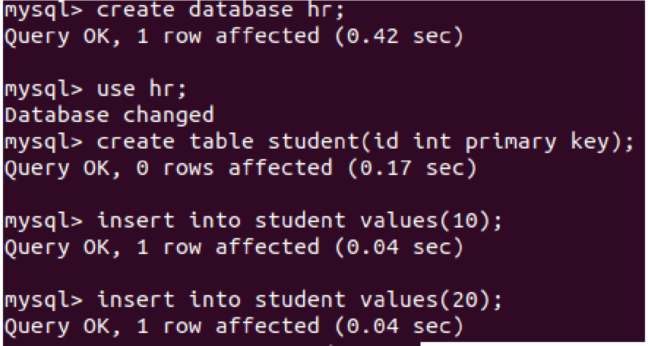
show databases;
create database hr;
use hr;
create table student(id int primary key);
insert into student values(10);
insert into student values(20);
select * from student;
quit;



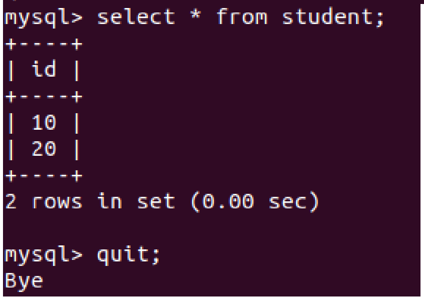
```
hduser@hduser-VirtualBox:/usr/local$ sudo service mysql restart
hduser@hduser-VirtualBox:/usr/local$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.02 sec)
```

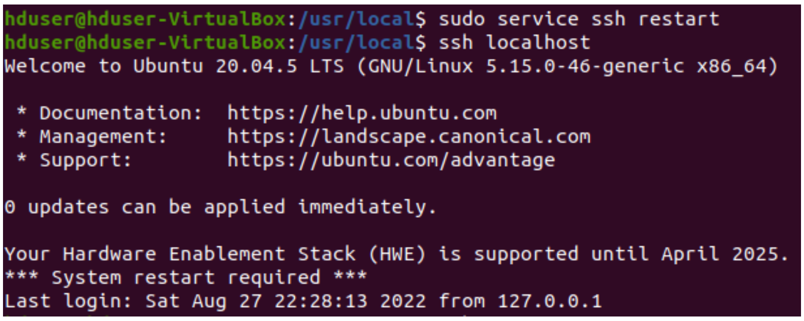


```
mysql> create database hr;
Query OK, 1 row affected (0.42 sec)

mysql> use hr;
Database changed
mysql> create table student(id int primary key);
Query OK, 0 rows affected (0.17 sec)

mysql> insert into student values(10);
Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(20);
Query OK, 1 row affected (0.04 sec)
```



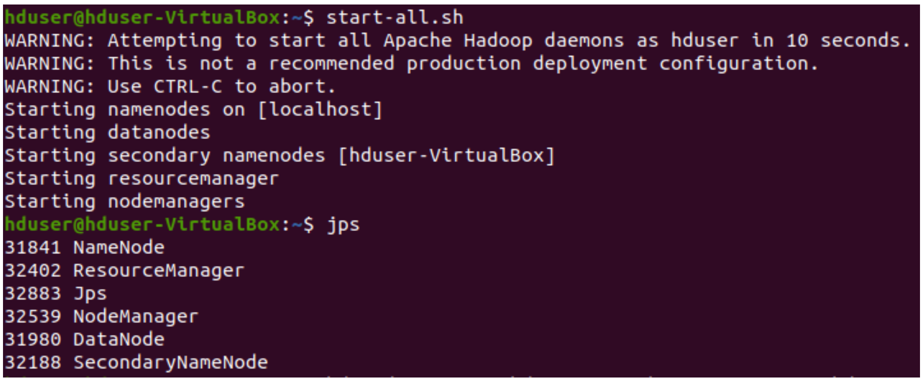
```
mysql> select * from student;
+----+
| id |
+----+
| 10 |
| 20 |
+----+
2 rows in set (0.00 sec)

mysql> quit;
Bye
```

2. START HADOOP:
   - ➢ sudo service ssh restart
   - ➢ ssh localhost



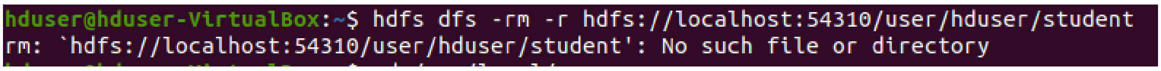
```
hduser@hduser-VirtualBox:/usr/local$ sudo service ssh restart
hduser@hduser-VirtualBox:/usr/local$ ssh localhost
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Sat Aug 27 22:28:13 2022 from 127.0.0.1
```

   - ➢ start-all.sh
   - ➢ jps



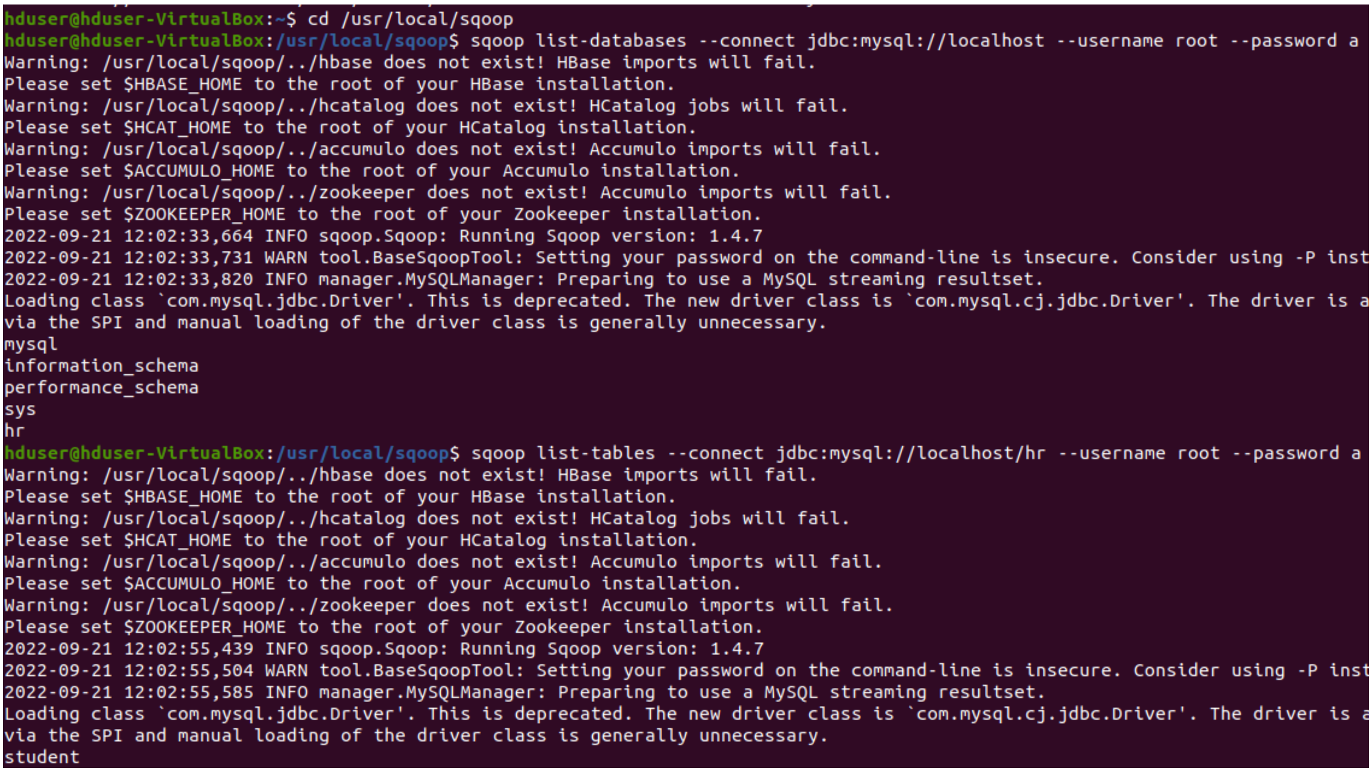
```
hduser@hduser-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [hduser-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hduser@hduser-VirtualBox:~$ jps
31841 NameNode
32402 ResourceManager
32883 Jps
32539 NodeManager
31980 DataNode
32188 SecondaryNameNode
```

- ➢ hdfs dfs -rm -r hdfs://localhost:54310/user/hduser/student

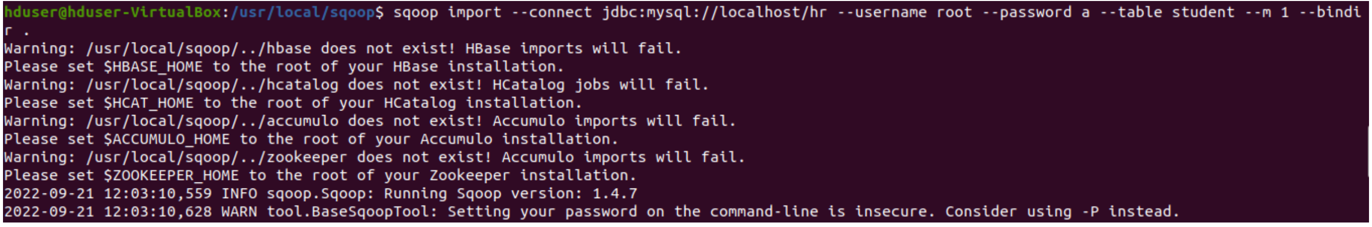


```
hduser@hduser-VirtualBox:~$ hdfs dfs -rm -r hdfs://localhost:54310/user/hduser/student
rm: `hdfs://localhost:54310/user/hduser/student': No such file or directory
```

3. Connect Sqoop to view databases and tables in MySQL:
   - ➢ cd /usr/local/sqoop
   - ➢ sqoop list-databases --connect jdbc:mysql://localhost --username root --password a
   - ➢ sqoop list-tables --connect jdbc:mysql://localhost/hr --username root --password a
   - ➢ sqoop import --connect jdbc:mysql://localhost/hr --username root --password a --table student --m 1 --bindir .

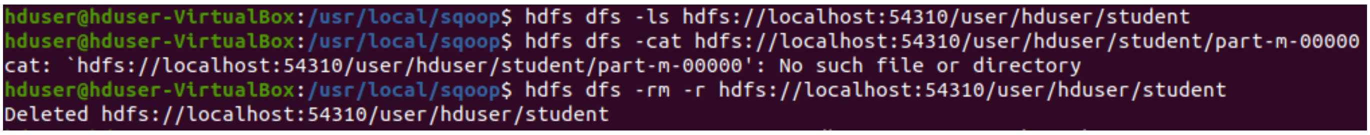


```
hduser@hduser-VirtualBox:~$ cd /usr/local/sqoop
hduser@hduser-VirtualBox:/usr/local/sqoop$ sqoop list-databases --connect jdbc:mysql://localhost --username root --password a
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2022-09-21 12:02:33,664 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2022-09-21 12:02:33,731 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P inst
2022-09-21 12:02:33,820 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is a
via the SPI and manual loading of the driver class is generally unnecessary.
mysql
information_schema
performance_schema
sys
hr
hduser@hduser-VirtualBox:/usr/local/sqoop$ sqoop list-tables --connect jdbc:mysql://localhost/hr --username root --password a
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2022-09-21 12:02:55,439 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2022-09-21 12:02:55,504 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P inst
2022-09-21 12:02:55,585 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is a
via the SPI and manual loading of the driver class is generally unnecessary.
student
```



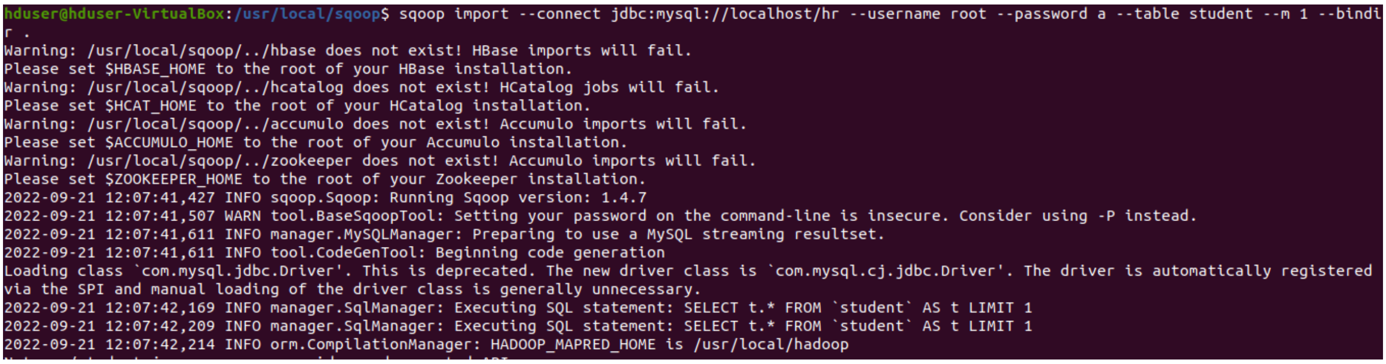
```
hduser@hduser-VirtualBox:/usr/local/sqoop$ sqoop import --connect jdbc:mysql://localhost/hr --username root --password a --table student --m 1 --bindi
r .
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2022-09-21 12:03:10,559 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2022-09-21 12:03:10,628 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
```

   - ➢ hdfs dfs -ls hdfs://localhost:54310/user/hduser/student
   - ➢ hdfs dfs -cat hdfs://localhost:54310/user/hduser/student/part-m-00000
   - ➢ hdfs dfs -rm -r hdfs://localhost:54310/user/hduser/student

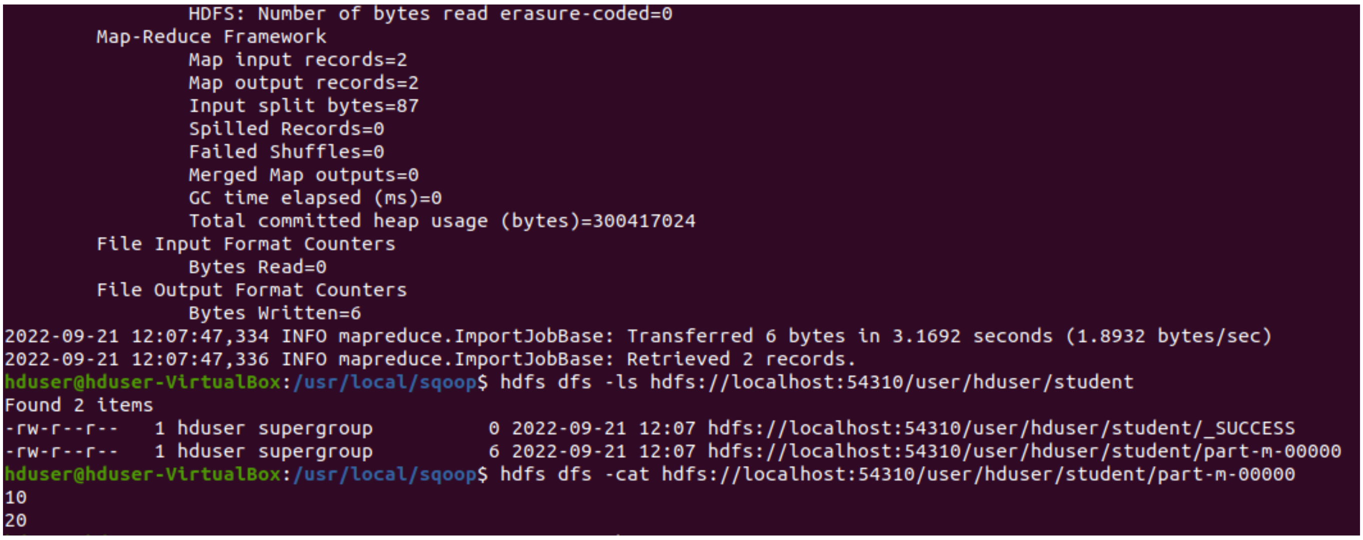


```
hduser@hduser-VirtualBox:/usr/local/sqoop$ hdfs dfs -ls hdfs://localhost:54310/user/hduser/student
hduser@hduser-VirtualBox:/usr/local/sqoop$ hdfs dfs -cat hdfs://localhost:54310/user/hduser/student/part-m-00000
cat: `hdfs://localhost:54310/user/hduser/student/part-m-00000': No such file or directory
hduser@hduser-VirtualBox:/usr/local/sqoop$ hdfs dfs -rm -r hdfs://localhost:54310/user/hduser/student
Deleted hdfs://localhost:54310/user/hduser/student
```

   - ➢ sqoop import --connect jdbc:mysql://localhost/hr --username root --password a --table student --m 1 --bindir .
   - ➢ hdfs dfs -ls hdfs://localhost:54310/user/hduser/student
   - ➢ hdfs dfs -cat hdfs://localhost:54310/user/hduser/student/part-m-00000

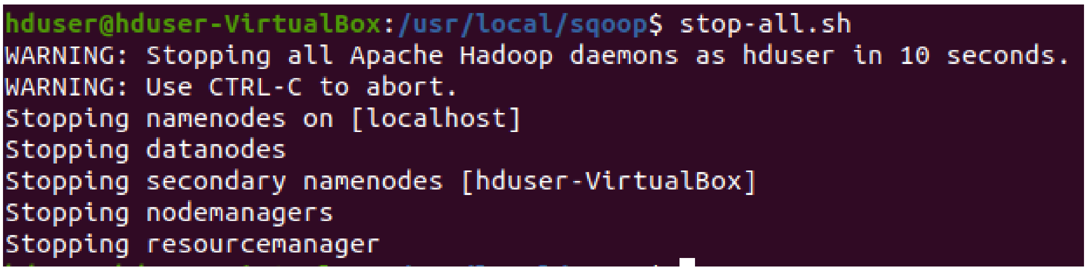


```
hduser@hduser-VirtualBox:/usr/local/sqoop$ sqoop import --connect jdbc:mysql://localhost/hr --username root --password a --table student --m 1 --bindi
r .
Warning: /usr/local/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2022-09-21 12:07:41,427 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2022-09-21 12:07:41,507 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2022-09-21 12:07:41,611 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2022-09-21 12:07:41,611 INFO tool.CodeGenTool: Beginning code generation
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered
via the SPI and manual loading of the driver class is generally unnecessary.
2022-09-21 12:07:42,169 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `student` AS t LIMIT 1
2022-09-21 12:07:42,209 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `student` AS t LIMIT 1
2022-09-21 12:07:42,214 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop
```



```
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=2
                Map output records=2
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=300417024
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=6
2022-09-21 12:07:47,334 INFO mapreduce.ImportJobBase: Transferred 6 bytes in 3.1692 seconds (1.8932 bytes/sec)
2022-09-21 12:07:47,336 INFO mapreduce.ImportJobBase: Retrieved 2 records.
hduser@hduser-VirtualBox:/usr/local/sqoop$ hdfs dfs -ls hdfs://localhost:54310/user/hduser/student
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-09-21 12:07 hdfs://localhost:54310/user/hduser/student/_SUCCESS
-rw-r--r--   1 hduser supergroup          6 2022-09-21 12:07 hdfs://localhost:54310/user/hduser/student/part-m-00000
hduser@hduser-VirtualBox:/usr/local/sqoop$ hdfs dfs -cat hdfs://localhost:54310/user/hduser/student/part-m-00000
10
20
```

4. STOP HADOOP:
   - ➢ stop-all.sh



```
hduser@hduser-VirtualBox:/usr/local/sqoop$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [hduser-VirtualBox]
Stopping nodemanagers
Stopping resourcemanager
```

Practical No: 8

Aim: Install and Configure Flume.

Theory:

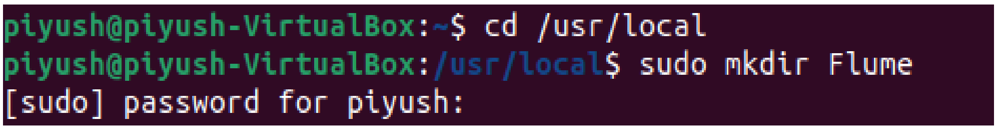
Apache Flume is a reliable and distributed system for collecting, aggregating and moving massive quantities of log data. It has a simple yet flexible architecture based on streaming data flows. Apache Flume is used to collect log data present in log files from web servers and aggregating it into HDFS for analysis.

Some Important features of FLUME:

- Flume has a flexible design based upon streaming data flows. It is fault tolerant and robust with multiple failovers and recovery mechanisms. Flume Big data has different levels of reliability to offer which includes 'best-effort delivery' and an 'end-to-end delivery'. Best-effort delivery does not tolerate any Flume node failure whereas 'end-to-end delivery' mode guarantees delivery even in the event of multiple node failures.
- Flume carries data between sources and sinks. This gathering of data can either be scheduled or event-driven. Flume has its own query processing engine which makes it easy to transform each new batch of data before it is moved to the intended sink.
- Possible Flume sinks include HDFS and HBase. Flume Hadoop can also be used to transport event data including but not limited to network traffic data, data generated by social media websites and email messages.

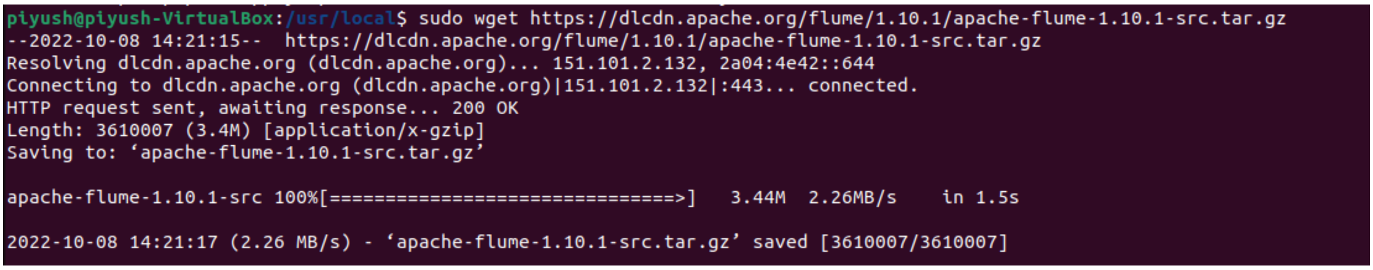
Steps (for Ubuntu VM):

1. download flume.
   - cd /usr/local
   - sudo mkdir Flume



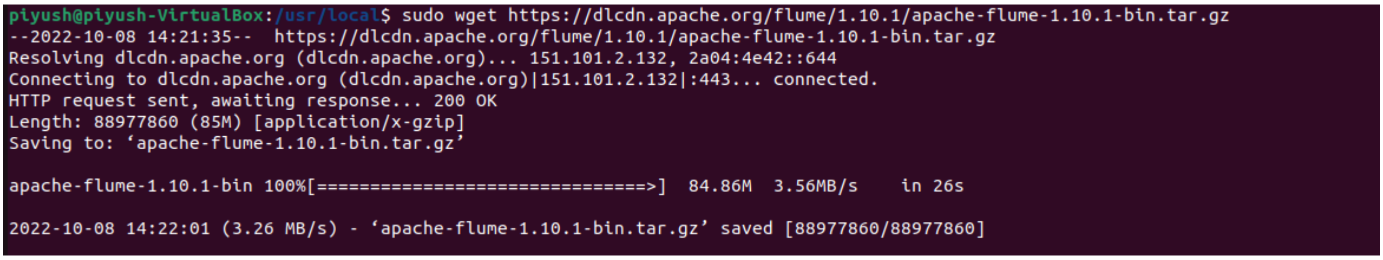
```
piyush@piyush-VirtualBox:~$ cd /usr/local
piyush@piyush-VirtualBox:/usr/local$ sudo mkdir Flume
[sudo] password for piyush:
```

   - sudo wget https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-src.tar.gz



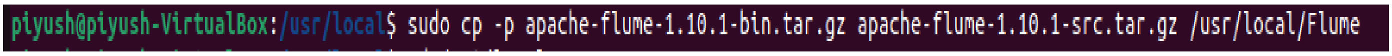
```
piyush@piyush-VirtualBox:/usr/local$ sudo wget https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-src.tar.gz
--2022-10-08 14:21:15--  https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-src.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3610007 (3.4M) [application/x-gzip]
Saving to: ‘apache-flume-1.10.1-src.tar.gz’

apache-flume-1.10.1-src 100%[===============================>]   3.44M  2.26MB/s    in 1.5s

2022-10-08 14:21:17 (2.26 MB/s) - ‘apache-flume-1.10.1-src.tar.gz’ saved [3610007/3610007]
```

   - sudo wget https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-bin.tar.gz



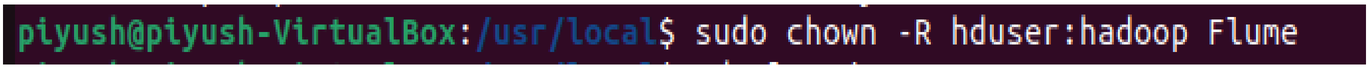
```
piyush@piyush-VirtualBox:/usr/local$ sudo wget https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-bin.tar.gz
--2022-10-08 14:21:35--  https://dlcdn.apache.org/flume/1.10.1/apache-flume-1.10.1-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 88977860 (85M) [application/x-gzip]
Saving to: ‘apache-flume-1.10.1-bin.tar.gz’

apache-flume-1.10.1-bin 100%[===============================>]  84.86M  3.56MB/s    in 26s

2022-10-08 14:22:01 (3.26 MB/s) - ‘apache-flume-1.10.1-bin.tar.gz’ saved [88977860/88977860]
```

2. open ubuntu and copy both files to Flume folder and change permission.
   - sudo cp -p apache-flume-1.10.1-bin.tar.gz apache-flume-1.10.1-src.tar.gz /usr/local/Flume



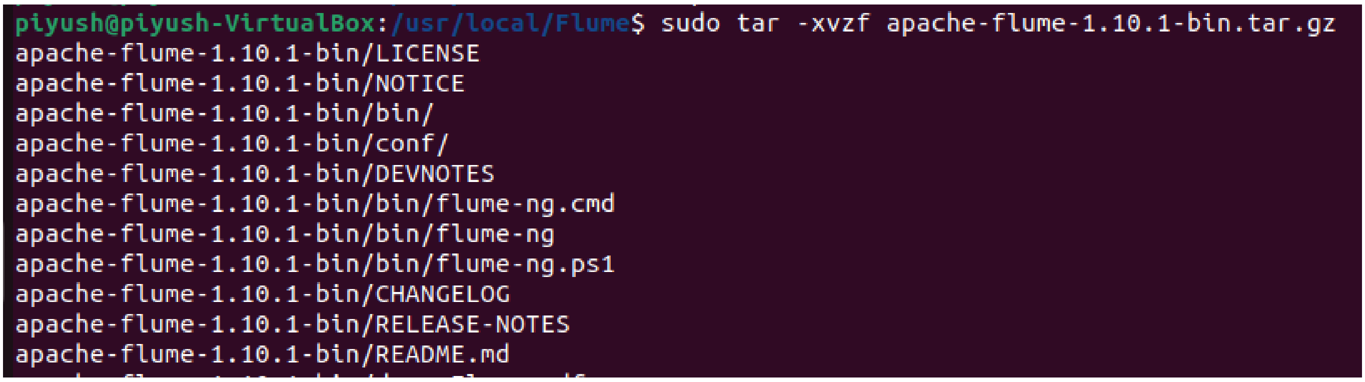
```
piyush@piyush-VirtualBox:/usr/local$ sudo cp -p apache-flume-1.10.1-bin.tar.gz apache-flume-1.10.1-src.tar.gz /usr/local/Flume
```

- ➢ sudo chown -R hduser:hadoop Flume



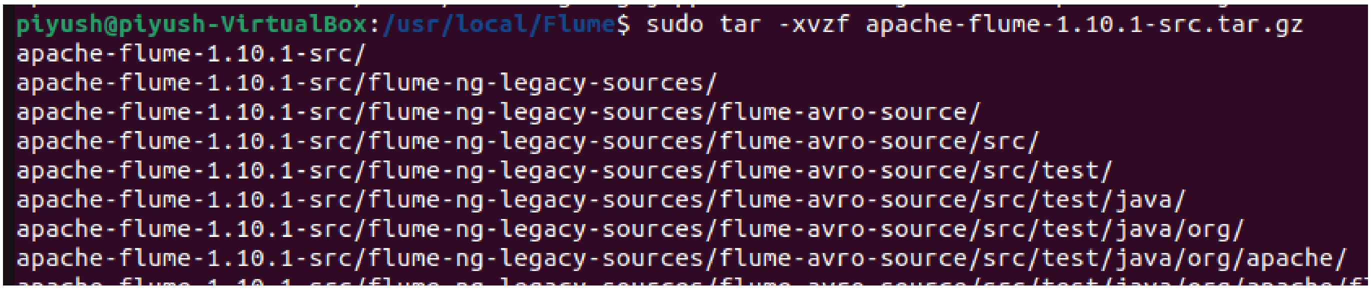
```
piyush@piyush-VirtualBox:/usr/local$ sudo chown -R hduser:hadoop Flume
```

3. untar and move it.
   - ➢ cd Flume/
   - ➢ sudo tar -xvzf apache-flume-1.10.1-bin.tar.gz



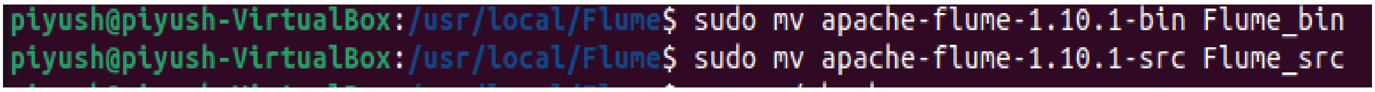
```
piyush@piyush-VirtualBox:/usr/local/Flume$ sudo tar -xvzf apache-flume-1.10.1-bin.tar.gz
apache-flume-1.10.1-bin/LICENSE
apache-flume-1.10.1-bin/NOTICE
apache-flume-1.10.1-bin/bin/
apache-flume-1.10.1-bin/conf/
apache-flume-1.10.1-bin/DEVNOTES
apache-flume-1.10.1-bin/bin/flume-ng.cmd
apache-flume-1.10.1-bin/bin/flume-ng
apache-flume-1.10.1-bin/bin/flume-ng.ps1
apache-flume-1.10.1-bin/CHANGELOG
apache-flume-1.10.1-bin/RELEASE-NOTES
apache-flume-1.10.1-bin/README.md
```

   - ➢ sudo tar -xvzf apache-flume-1.10.1-src.tar.gz



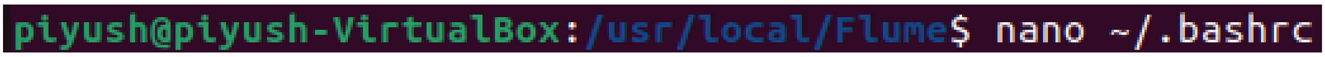
```
piyush@piyush-VirtualBox:/usr/local/Flume$ sudo tar -xvzf apache-flume-1.10.1-src.tar.gz
apache-flume-1.10.1-src/
apache-flume-1.10.1-src/flume-ng-legacy-sources/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/src/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/src/test/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/src/test/java/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/src/test/java/org/
apache-flume-1.10.1-src/flume-ng-legacy-sources/flume-avro-source/src/test/java/org/apache/
```

   - ➢ sudo mv apache-flume-1.10.1-bin Flume_bin
   - ➢ sudo mv apache-flume-1.10.1-src Flume_src



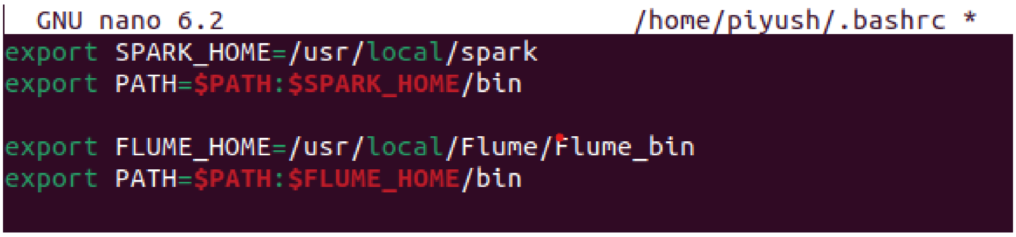
```
piyush@piyush-VirtualBox:/usr/local/Flume$ sudo mv apache-flume-1.10.1-bin Flume_bin
piyush@piyush-VirtualBox:/usr/local/Flume$ sudo mv apache-flume-1.10.1-src Flume_src
```

4. Add FLUME_HOME path to bashrc
   - ➢ nano ~/.bashrc



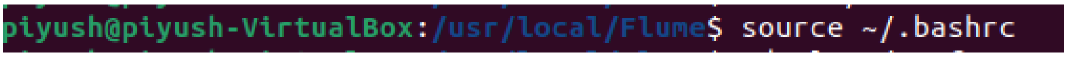
```
piyush@piyush-VirtualBox:/usr/local/Flume$ nano ~/.bashrc
```

export FLUME_HOME=/usr/local/Flume/Flume_bin
export PATH=$PATH:$FLUME_HOME/bin



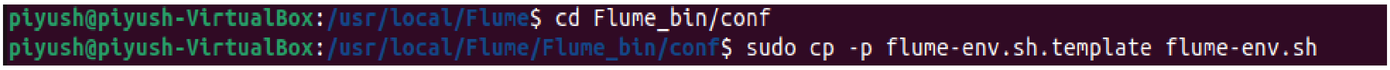
```
  GNU nano 6.2                    /home/piyush/.bashrc *
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin

export FLUME_HOME=/usr/local/Flume/Flume_bin
export PATH=$PATH:$FLUME_HOME/bin
```

   - ➢ source ~/.bashrc



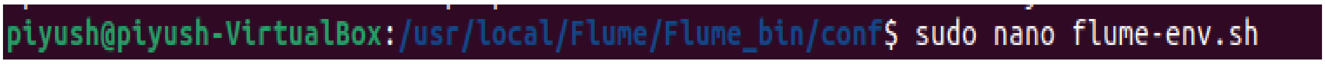
```
piyush@piyush-VirtualBox:/usr/local/Flume$ source ~/.bashrc
```

5. copy files.
   - ➢ cd Flume_bin/conf
   - ➢ sudo cp -p flume-env.sh.template flume-env.sh



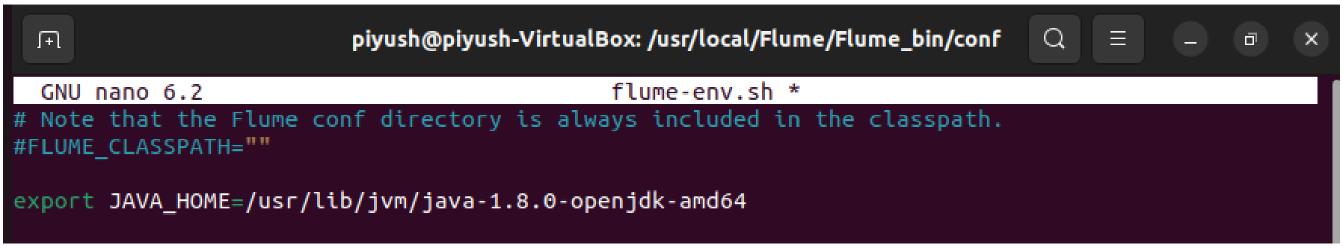
```
piyush@piyush-VirtualBox:/usr/local/Flume$ cd Flume_bin/conf
piyush@piyush-VirtualBox:/usr/local/Flume/Flume_bin/conf$ sudo cp -p flume-env.sh.template flume-env.sh
```

6. open flume-env.sh and add java path JAVA_HOME :
   - ➢ sudo nano flume-env.sh



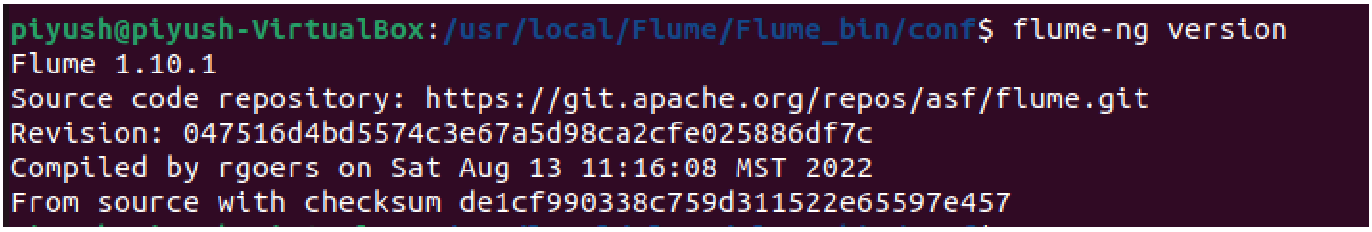
```
piyush@piyush-VirtualBox:/usr/local/Flume/Flume_bin/conf$ sudo nano flume-env.sh
```

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64



```
  GNU nano 6.2                          flume-env.sh *
# Note that the Flume conf directory is always included in the classpath.
#FLUME_CLASSPATH=""

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

7. type flume-ng you should get desired output
   - ➢ flume-ng version



```
piyush@piyush-VirtualBox:/usr/local/Flume/Flume_bin/conf$ flume-ng version
Flume 1.10.1
Source code repository: https://git.apache.org/repos/asf/flume.git
Revision: 047516d4bd5574c3e67a5d98ca2cfe025886df7c
Compiled by rgoers on Sat Aug 13 11:16:08 MST 2022
From source with checksum de1cf990338c759d311522e65597e457
```