EXPERIMENT 2

NAME	**Shreya Shetty**
UID	**2019140059**
CLASS	**TE IT**
BATCH	**A**
SUBJECT	**Big Data Analytics Lab**

AIM: Installation of Hadoop and execution of HDFS commands

HADOOP INSTALLATION:

1. Check Java Version

```
shruti@shruti-VirtualBox:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

2. Install SSH

```
shruti@shruti-VirtualBox:~$ sudo apt install openssh-server openssh-client -
[sudo] password for shruti:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:8.2p1-4ubuntu0.4).
openssh-client set to manually installed.
openssh-server is already the newest version (1:8.2p1-4ubuntu0.4).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 193 not upgraded.
```

3. Generating SSH key pair and storing : ssh-keygen -t rsa -P " -f ~/.ssh/id_rsa

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/shruti/.ssh/id_rsa):
/home/shruti/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/shruti/.ssh/id_rsa
Your public key has been saved in /home/shruti/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:D5PRujnVk5Hb1au48+CrjfxdOoZcx8d7PcSowW8fIBM shruti@shruti
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|          .   .  .|
|         . .Eo   o|
|          + ..= ..|
|         S oo=.=o |
|          B o++o=o|
|         + oo*.oo+|
|         ..+=+=oo+|
|          +o**+o.o|
+----[SHA256]-----+
```

4. Ssh localhost

```
shruti@shruti-VirtualBox:~$ ssh localhost
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-30-generic x86_64)

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

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

Your Hardware Enablement Stack (HWE) is supported until April 2025.
```

5. Edit the .bashrc shell configuration file using a text editor

```
shruti@shruti-VirtualBox:~$ gedit ~/.bashrc
```

Open | .bashrc | Save

```
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 #Hadoop Related Options
120 export HADOOP_HOME=/home/shruti/hadoop
121 export HADOOP_INSTALL=$HADOOP_HOME
122 export HADOOP_MAPRED_HOME=$HADOOP_HOME
123 export HADOOP_COMMON_HOME=$HADOOP_HOME
124 export HADOOP_HDFS_HOME=$HADOOP_HOME
125 export YARN_HOME=$HADOOP_HOME
126 export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
127 export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
128 export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
129
130
```

6. Specifying which Java implementation is to be utilized

GNU nano 4.8 /home/hdoop/hadoop-3.2.2/etc/hadoop/hadoop-env.sh Modified

```
###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
# export JAVA_HOME=
export JAVA_HOME = /usr/lib/jvm/java-8-openjdk-amd64

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

7. Making two folders namenode and datanode

```
sudo mkdir -p /home/shruti/hadoop/store/hdfs/namenode
sudo mkdir -p /home/shruti/hadoop/store/hdfs/datanode
```

8. Opening the following files in text editor and adding commands between the conf tags

   a. core-site.xml

core-site.xml | mapred-site.xml | hdfs-site.xml | yarn-site.xml

```
  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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/amrit/hadoop-3.1.4/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>
```

b. mapred-site.xml

core-site.xml | **mapred-site.xml** | hdfs-site.xml | yarn-site.xml

```
<!--
  Licensed 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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<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>
```

c. hdfs-site.xml

core-site.xml | mapred-site.xml | **hdfs-site.xml** | yarn-site.xml

```
    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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

d. yarn-site.xml

core-site.xml | mapred-site.xml | hdfs-site.xml | yarn-site.xml

```
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>127.0.0.1</value>
</property>
<property>
<name>yarn.acl.enable</name>
<value>0</value>
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPA
value>
</property>
```

9. Formatting the NameNode before starting Hadoop services for the first time

```
shruti@shruti-VirtualBox:~$ hdfs namenode -format
2022-03-05 15:42:20,846 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = shruti-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.2
STARTUP_MSG:   classpath = /home/shruti/hadoop/etc/hadoop:/home/shruti/hadoop/s
hare/hadoop/common/lib/commons-codec-1.11.jar:/home/shruti/hadoop/share/hadoop/
common/lib/httpcore-4.4.13.jar:/home/shruti/hadoop/share/hadoop/common/lib/comm
ons-configuration2-2.1.1.jar:/home/shruti/hadoop/share/hadoop/common/lib/jersey
-server-1.19.jar:/home/shruti/hadoop/share/hadoop/common/lib/commons-compress-1
.19.jar:/home/shruti/hadoop/share/hadoop/common/lib/jackson-databind-2.9.10.4.j
ar:/home/shruti/hadoop/share/hadoop/common/lib/error_prone_annotations-2.2.0.ja
r:/home/shruti/hadoop/share/hadoop/common/lib/commons-lang3-3.7.jar:/home/shrut
i/hadoop/share/hadoop/common/lib/checker-qual-2.5.2.jar:/home/shruti/hadoop/sha
re/hadoop/common/lib/re2j-1.1.jar:/home/shruti/hadoop/share/hadoop/common/lib/h
adoop-auth-3.2.2.jar:/home/shruti/hadoop/share/hadoop/common/lib/jaxb-api-2.2.1
1.jar:/home/shruti/hadoop/share/hadoop/common/lib/kerby-util-1.0.1.jar:/home/sh
ruti/hadoop/share/hadoop/common/lib/avro-1.7.7.jar:/home/shruti/hadoop/share/ha
doop/common/lib/jetty-xml-9.4.20.v20190813.jar:/home/shruti/hadoop/share/hadoop
/common/lib/audience-annotations-0.5.0.jar:/home/shruti/hadoop/share/hadoop/com
mon/lib/jetty-servlet-9.4.20.v20190813.jar:/home/shruti/hadoop/share/hadoop/com
```

10. Starting the Hadoop services using the following command

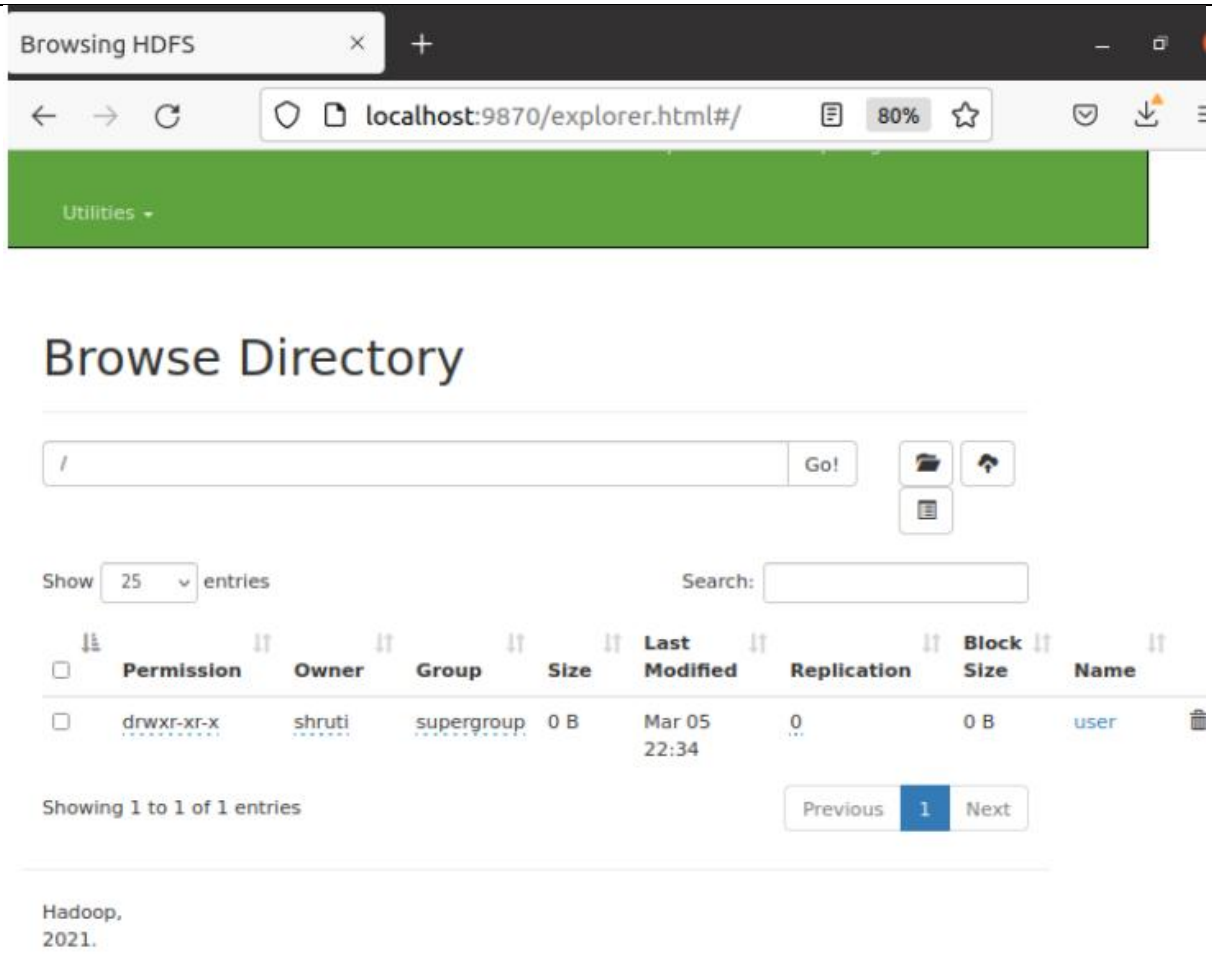
```
shruti@shruti-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as shruti in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [shruti-VirtualBox]
shruti-VirtualBox: Warning: Permanently added 'shruti-virtualbox' (ECDSA) to th
e list of known hosts.
2022-03-05 15:42:52,069 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
```

11. Checking if the Hadoop services are up and running

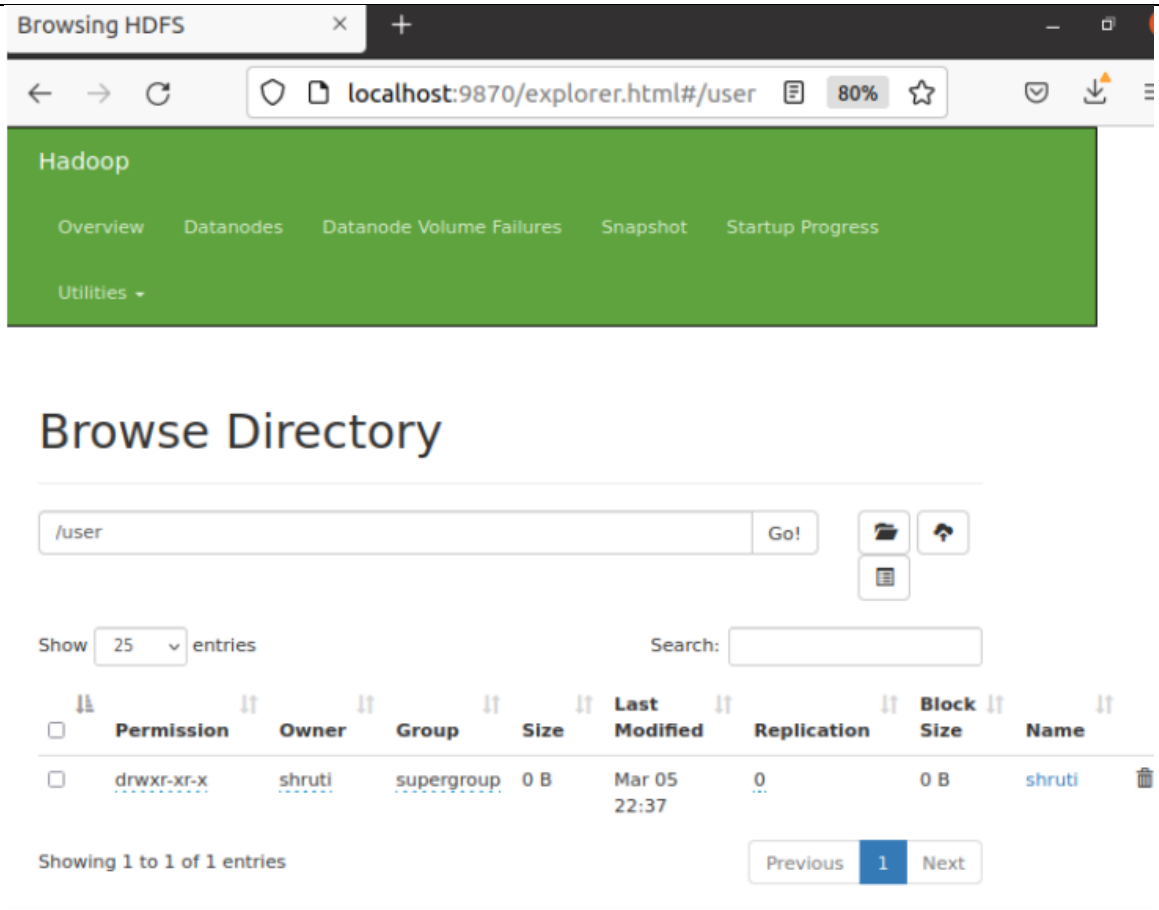
```
shruti@shruti-VirtualBox:~$ jps
9122 NameNode
9254 DataNode
9767 NodeManager
10073 Jps
9628 ResourceManager
```

HDFS COMMANDS:

1	**ls:** This command is used to list all the files. Use *lsr* for recursive approach. It is useful when we want a hierarchy of a folder.
	
	Syntax:
	
	bin/hdfs dfs -ls <path>
	
	`shruti@shruti-VirtualBox:~$ hdfs dfs -ls /`
	`2022-03-05 22:35:21,092 WARN util.NativeCodeLoader: Unable to load native-hadoo`
	`p library for your platform... using builtin-java classes where applicable`
	`Found 1 items`
	`drwxr-xr-x - shruti supergroup 0 2022-03-05 22:34 /user`
	`shruti@shruti-VirtualBox:~$`



2	**mkdir**: To create a directory. In Hadoop *dfs* there is no home directory by default.
	Syntax:
	bin/hdfs dfs -mkdir <folder name>
	creating home directory:
	hdfs/bin -mkdir /user
	hdfs/bin -mkdir /user/username -> write the username of your
	`shruti@shruti-VirtualBox:~$ hdfs dfs -mkdir /user/shruti`
	`2022-03-05 22:37:12,248 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable`
	`shruti@shruti-VirtualBox:~$`



3 **touchz**: It creates an empty file.

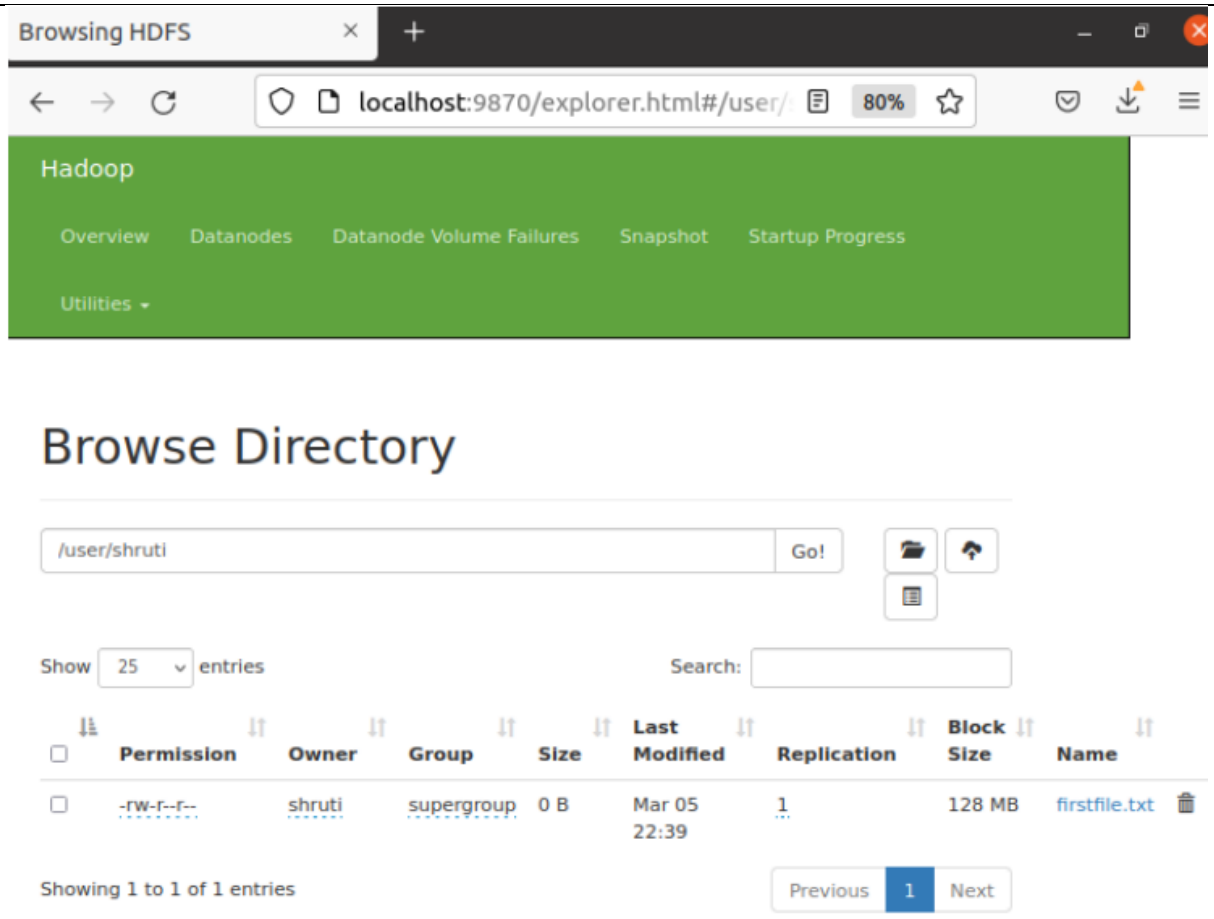
Syntax:

bin/hdfs dfs -touchz <file_path>

Example:

bin/hdfs dfs -touchz /rupali/myfile.txt

```
shruti@shruti-VirtualBox:~$ hdfs dfs -touchz /user/shruti/firstfile.txt
2022-03-05 22:39:23,941 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
```



4 **copyFromLocal (or) put:** To copy files/folders from local file system to hdfs store.

Syntax:

bin/hdfs dfs -copyFromLocal <local file path> <dest(present on hdfs)>

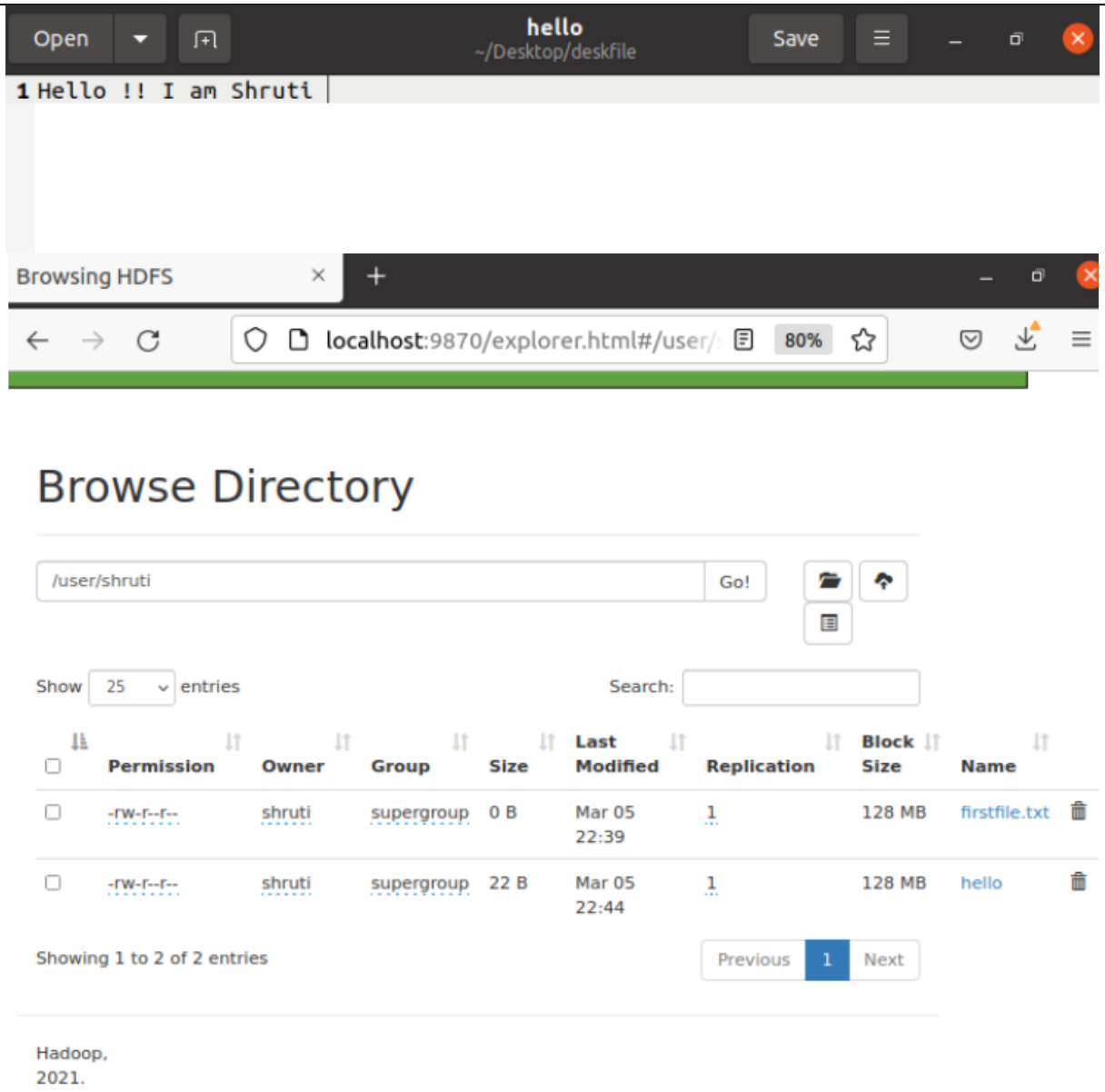
Example: Let's suppose we have a file *AI.txt* on Desktop which we want to copy to folder *Bigdata* present on hdfs.

bin/hdfs dfs -copyFromLocal ../Desktop/AI.txt /Bigdata

(OR)

bin/hdfs dfs -put ../Desktop/AI.txt /Bigdata

```
shruti@shruti-VirtualBox:~$ hdfs dfs -copyFromLocal /home/shruti/Desktop/deskfi
le/hello /user/shruti
2022-03-05 22:44:23,473 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
```



5

cat: To print file contents.

Syntax:

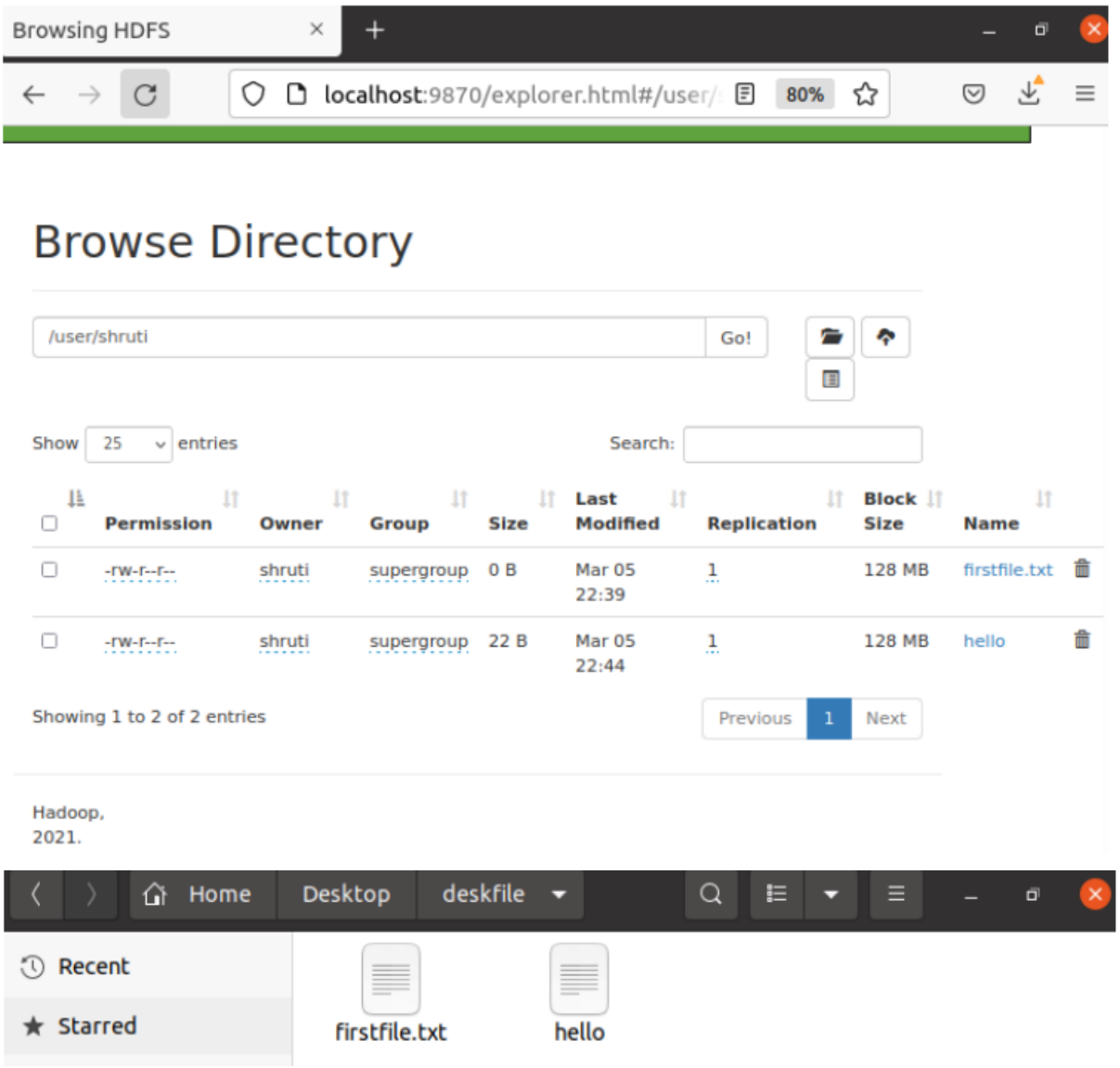
bin/hdfs dfs -cat <path>

Example:

// print the content of AI.txt present

bin/hdfs dfs -cat /Bigdata/AI.txt

```
shruti@shruti-VirtualBox:~$ hdfs dfs -cat /user/shruti/hello
2022-03-05 22:46:58,277 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Hello !! I am Shruti
```

6	**copyToLocal (or) get:** To copy files/folders from hdfs store to local file system.
	
	Syntax:
	
	bin/hdfs dfs -copyToLocal <<srcfile(on hdfs)> <local file dest>
	
	Example:
	bin/hdfs dfs -copyToLocal /Bigdata ../Desktop/rupali
	
	(OR)
	
	bin/hdfs dfs -get /geeks/myfile.txt ../Desktop/rupali
	
	`shruti@shruti-VirtualBox:~$ hdfs dfs -copyToLocal /user/shruti/firstfile.txt /home/shruti/Desktop/deskfile/`
	`2022-03-05 22:49:24,487 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable`
	
	 

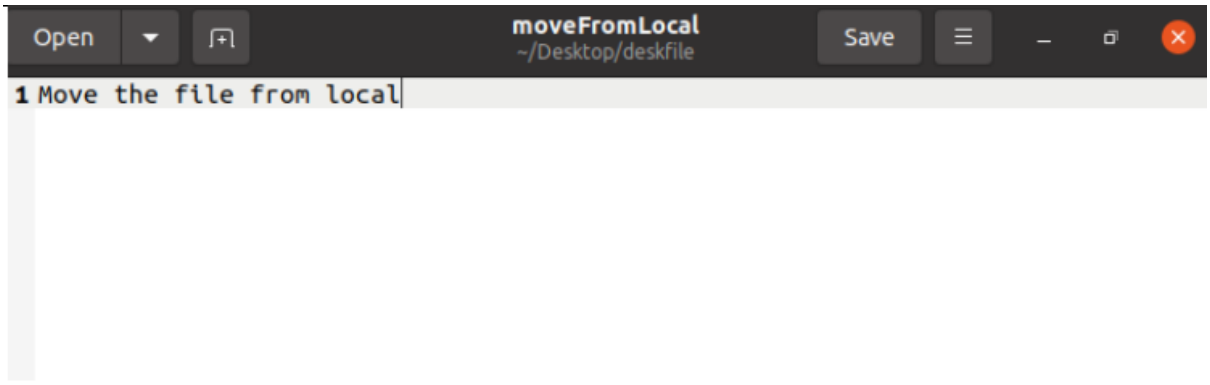
7	**moveFromLocal:** This command will move file from local to hdfs.

Syntax:

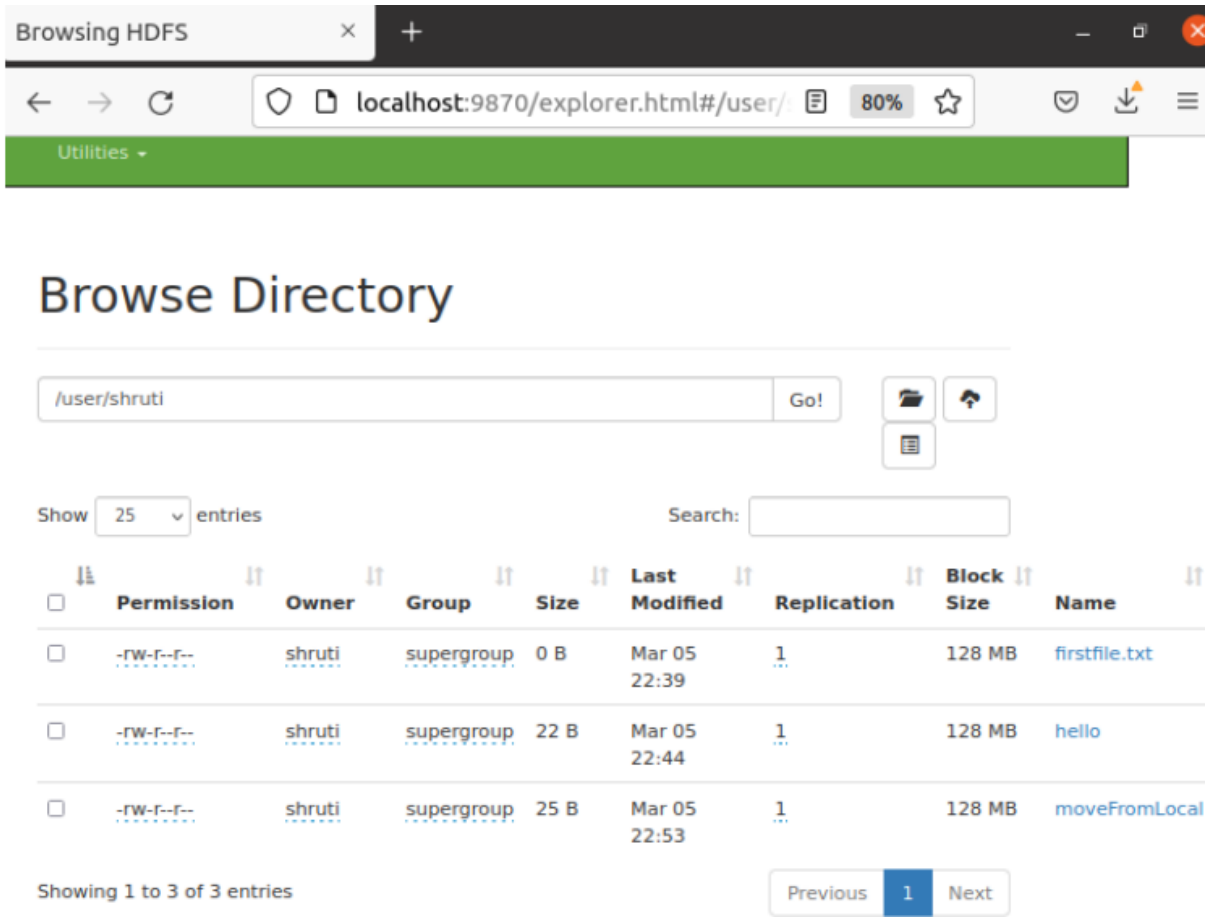
bin/hdfs dfs -moveFromLocal <local src> <dest(on hdfs)>

Example:

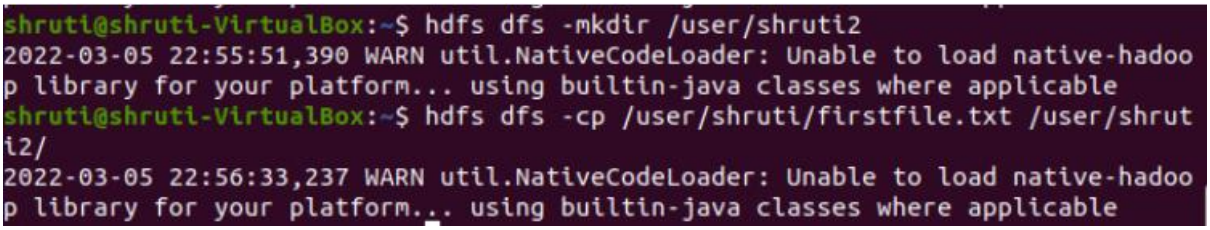
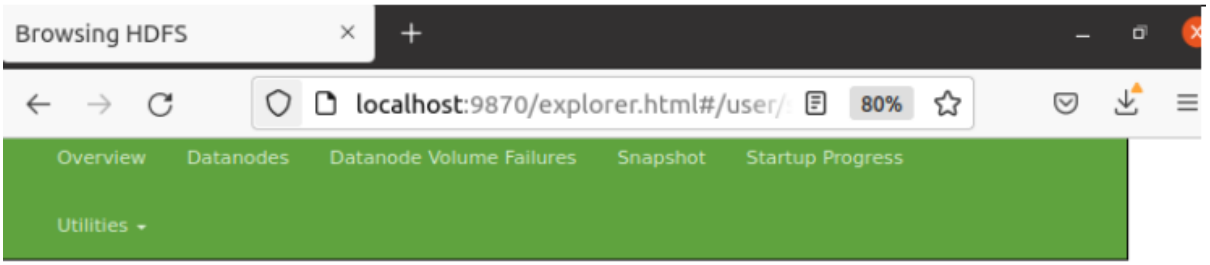
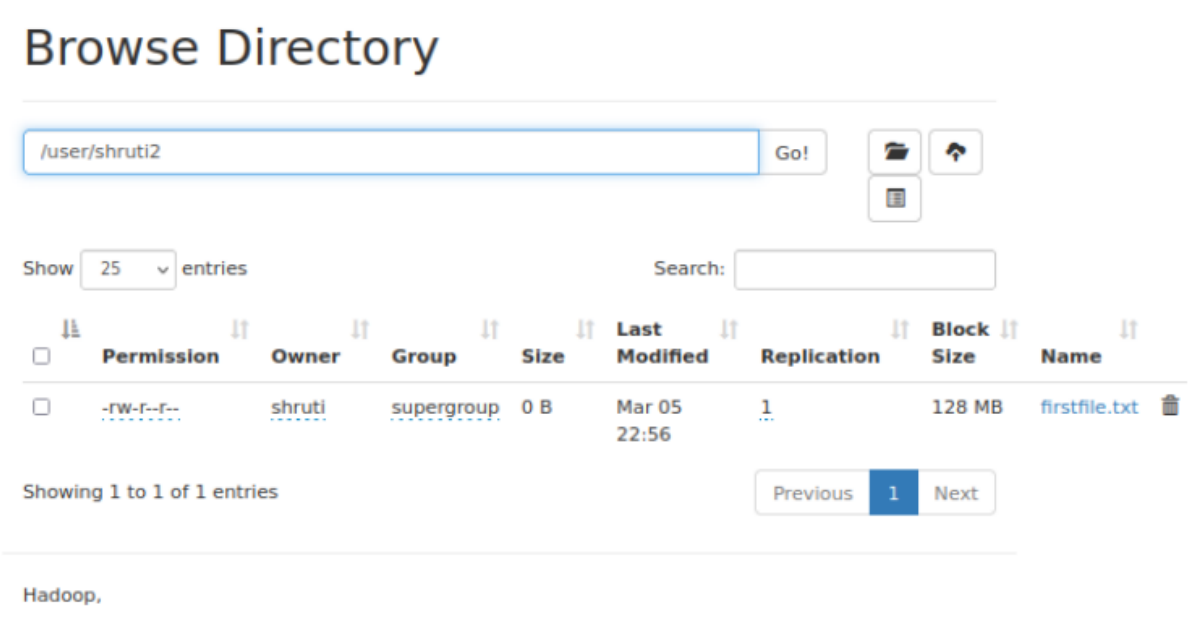
bin/hdfs dfs -moveFromLocal ../Desktop/cutAndPaste.txt /rupali




```
shruti@shruti-VirtualBox:~$ hdfs dfs -moveFromLocal /home/shruti/Desktop/deskfi
le/moveFromLocal /user/shruti
2022-03-05 22:53:29,391 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
```



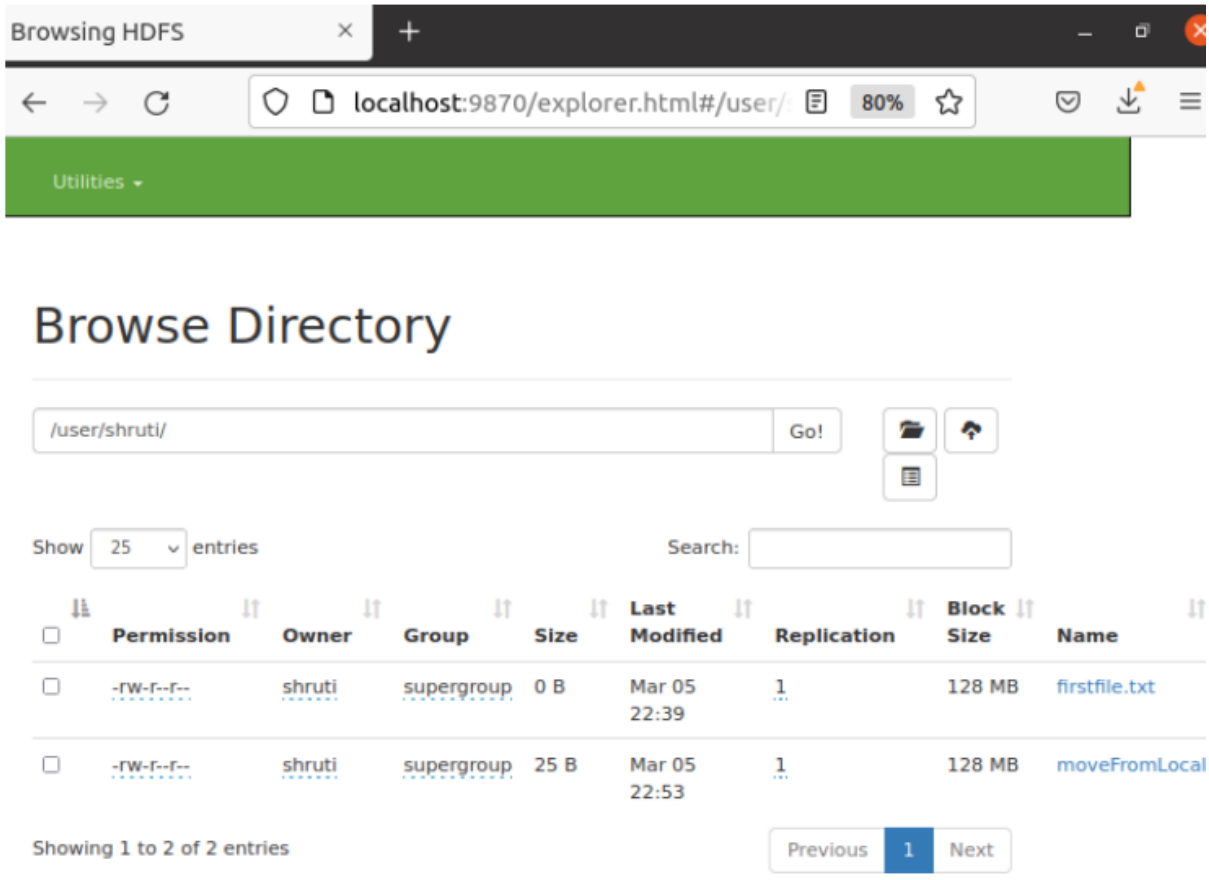
8 **cp:** This command is used to copy files within hdfs. Lets copy

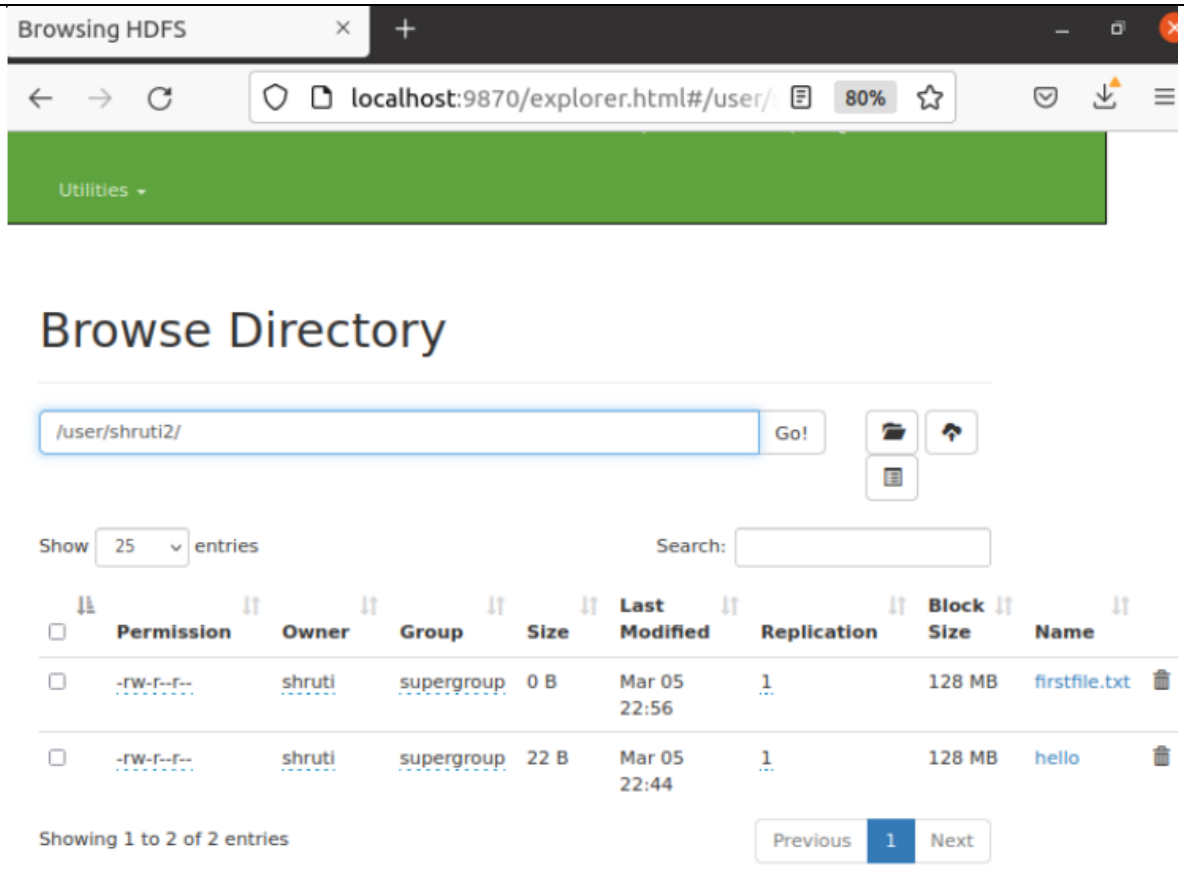
	folder *rupali* to *rupali_copied.* **Syntax:** bin/hdfs dfs -cp <src(on hdfs)> <dest(on hdfs)> **Example:** bin/hdfs -cp /rupali /rupali_copied   
9	**mv:** This command is used to move files within hdfs. Lets cut-paste a file *myfile.txt* from *rupali* folder to *rupali_copied.* **Syntax:** bin/hdfs dfs -mv <src(on hdfs)> <src(on hdfs)>

Example:

bin/hdfs -mv /rupali/myfile.txt /rupali_copied

```
shruti@shruti-VirtualBox:~$ hdfs dfs -mv /user/shruti/hello /user/shruti2/
2022-03-05 22:58:19,238 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
```





10 **rmr:** This command deletes a file from HDFS *recursively*. It is very useful command when you want to delete a *non-empty directory*.

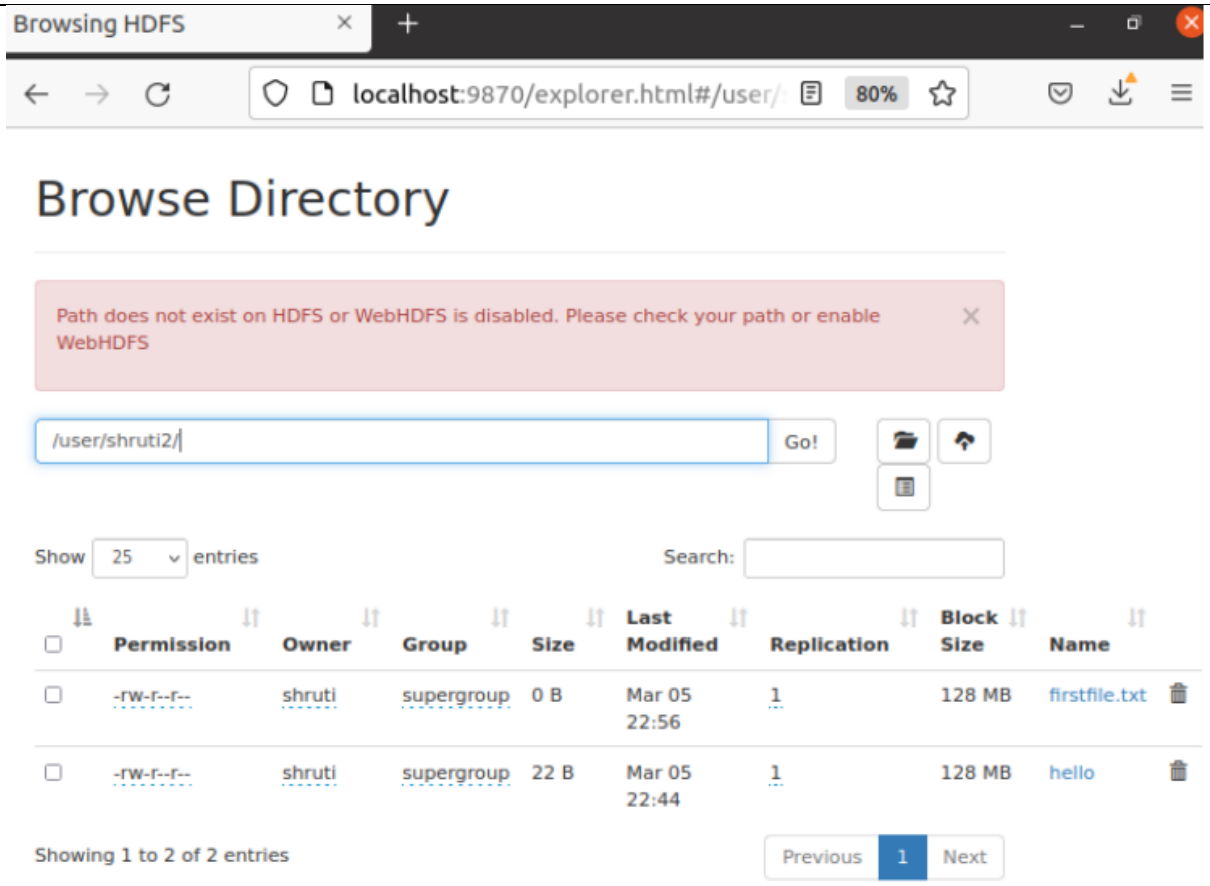
Syntax:

bin/hdfs dfs -rmr <filename/directoryName>

Example:

bin/hdfs dfs -rmr /rupali_copied -> It will delete all the content inside the directory then the directory itself.

```
shruti@shruti-VirtualBox:~$ hdfs dfs -rm -r /user/shruti2
2022-03-05 23:00:53,512 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Deleted /user/shruti2
```



11 **du:** It will give the size of each file in directory.

Syntax:

bin/hdfs dfs -du <dirName>

Example:

bin/hdfs dfs -du /rupali

```
shruti@shruti-VirtualBox:~$ hdfs dfs -du /user/shruti
2022-03-05 23:03:17,909 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
0   0   /user/shruti/firstfile.txt
25  25  /user/shruti/moveFromLocal
```

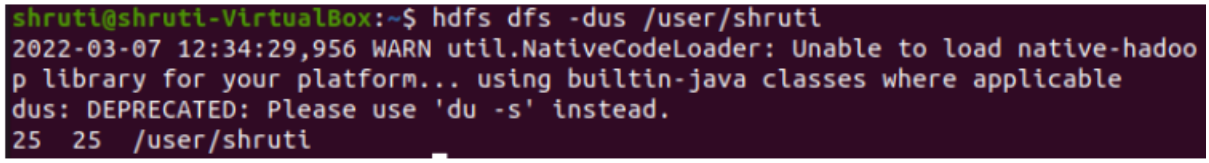
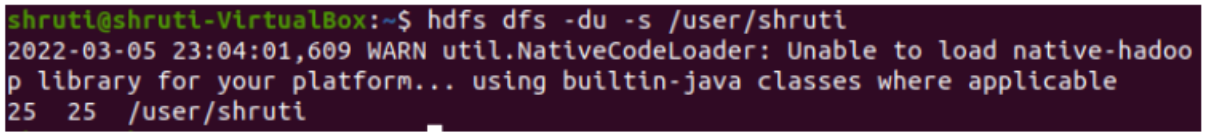
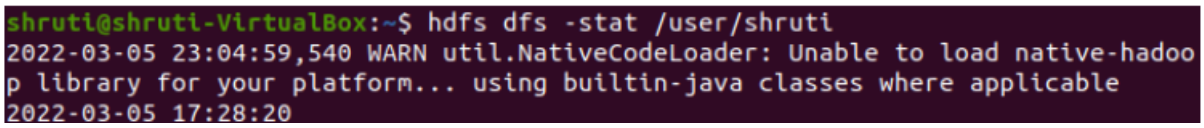
12 **dus:**: This command will give the total size of directory/file.

Syntax:

bin/hdfs dfs -dus <dirName>

Example:

bin/hdfs dfs -dus /rupali

	 
13	**stat:** It will give the last modified time of directory or path. In short it will give stats of the directory or file. **Syntax:** bin/hdfs dfs -stat <hdfs file> **Example:** bin/hdfs dfs -stat /rupali 

CONCLUSION:

In this experiment I installed Apache Hadoop and performed the given HDFS commands. Before starting with the HDFS command, we have to start the Hadoop services. 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 HDFS is a distributed file system that provides redundant storage space for files having huge sizes. It is used for storing files that are in the range of terabytes to petabytes. With the help of the HDFS command, we can perform Hadoop HDFS file operations like changing the file permissions, viewing the file contents, creating files or directories, copying file/directory from the local file system to HDFS or vice-versa, etc.