PRACTICAL NO :01

SET UP AND CONFIGURATION HADOOP USING CLOUDERA CREATING A HDFS SYSTEM WITH MINIMUM 1 NAME NODE AND 1 DATA NODES HDFS COMMANDS

Unit Structure :

1.1 OBJECTIVES

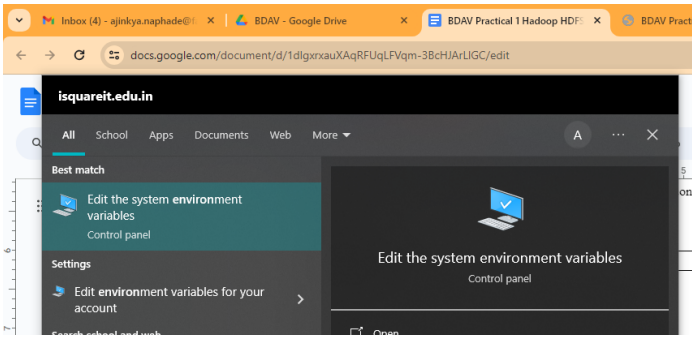
The Hadoop file system stores the data in multiple copies. Also, it's a cost effective solution for any business to store their data efficiently. HDFS Operations acts as the key to open the vaults in which you store the data to be available from remote locations. This chapter describes how to set up and edit the deployment configuration files for HDFS

1.2 PREREQUISITE

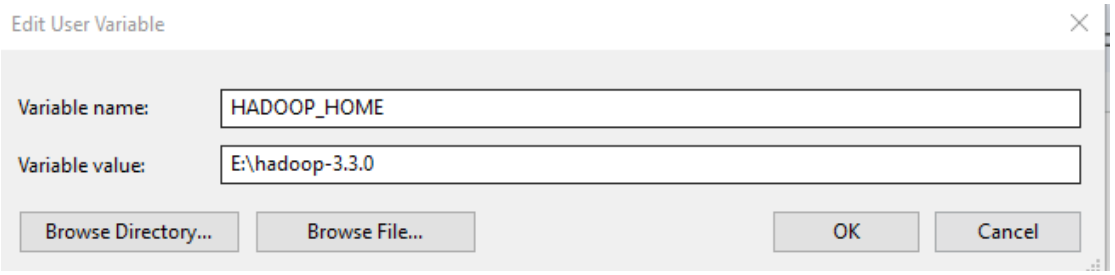
Check your java version through this command on command prompt.

java -version

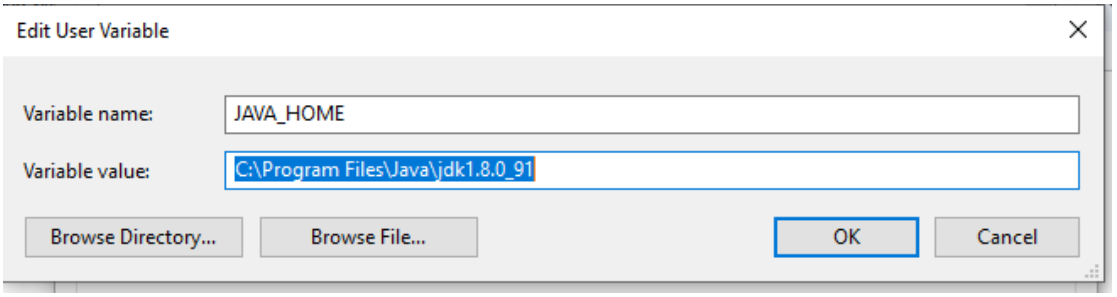
Create a new user variable. Put the Variable_name as HADOOP_HOME and Variable_value as the path of the bin folder where you extracted hadoop.



Enter administrative details as per need.

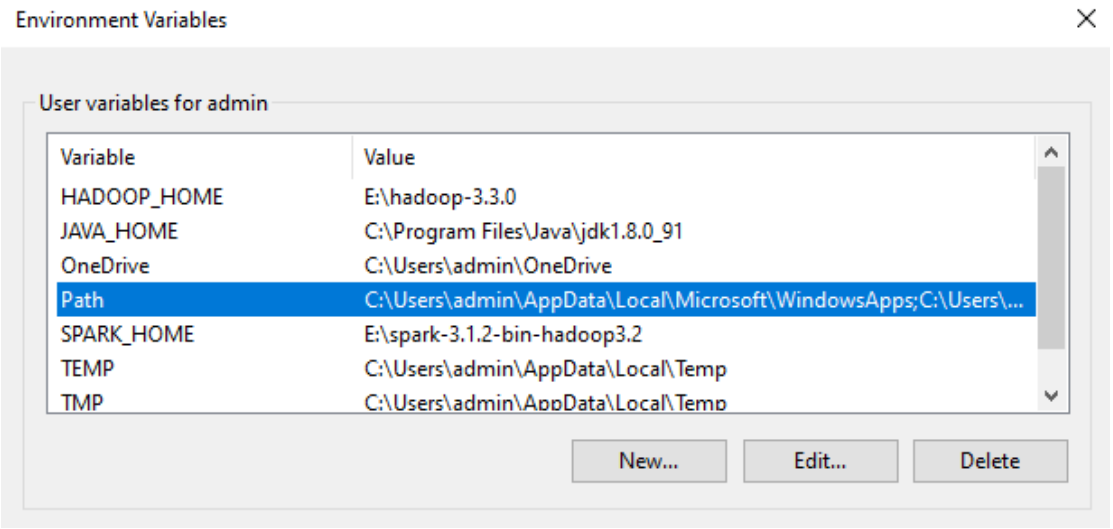


Likewise, create a new user variable with variable name as JAVA_HOME and variable value as the path of the bin folder in the Java directory.

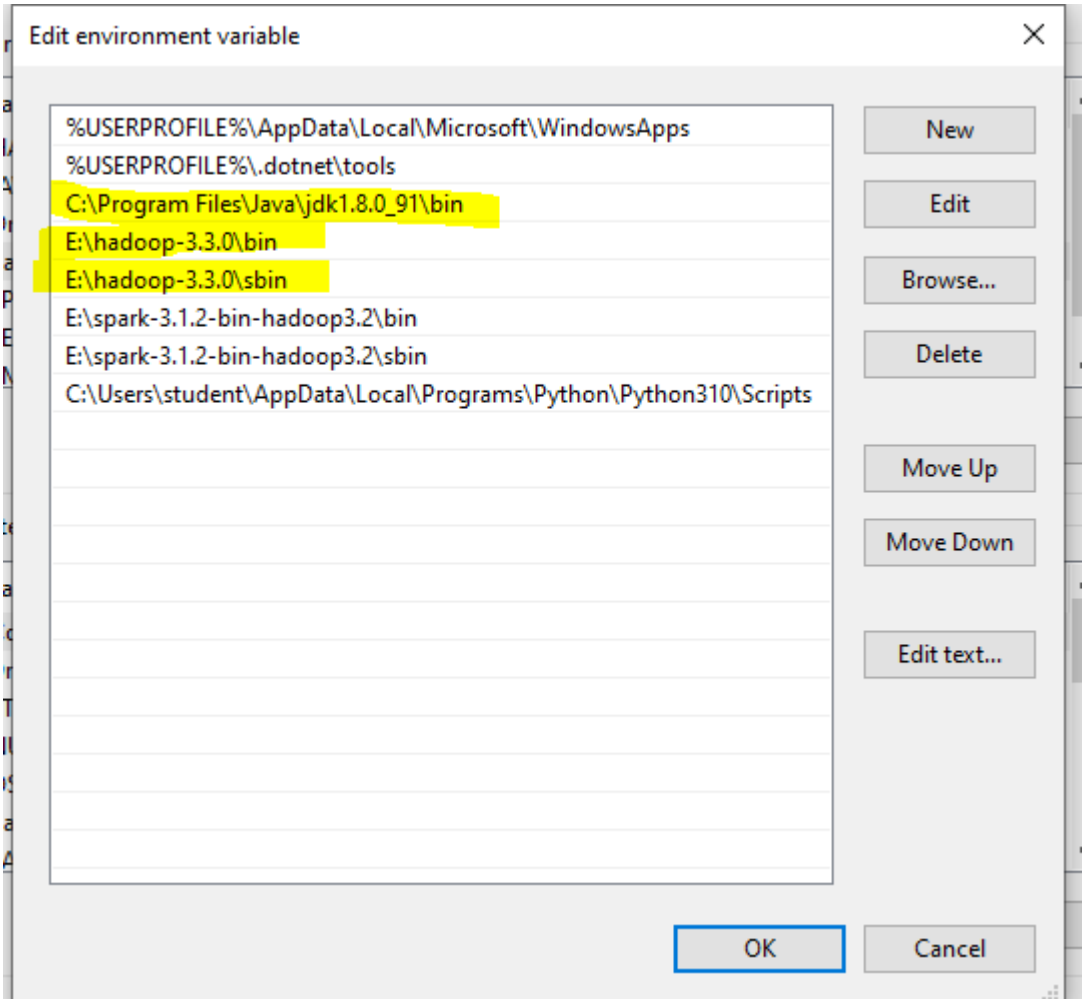


Now we need to set Hadoop bin directory and Java bin directory path in system variable path.

Edit Path in system variable :

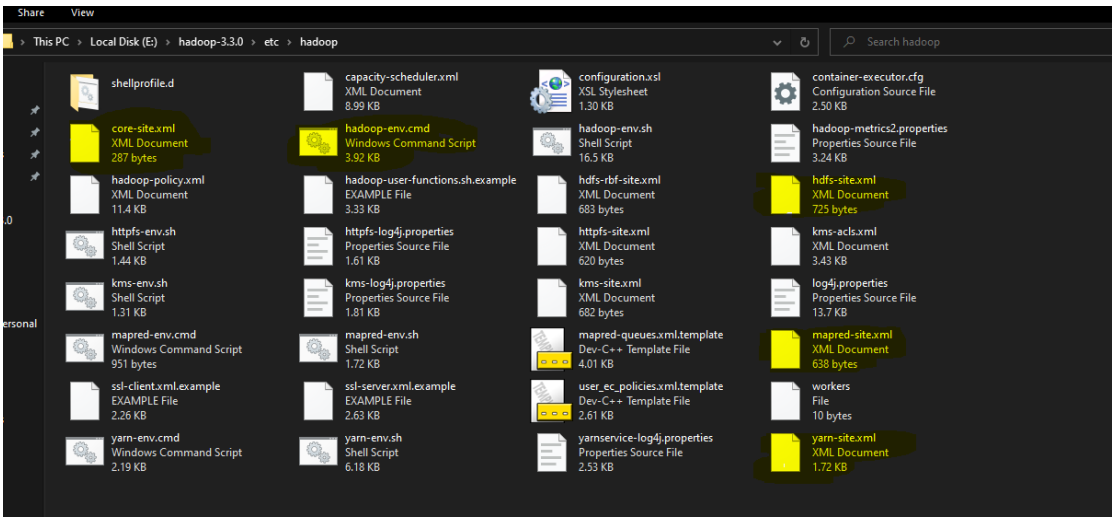


Click on New and add the bin directory path of Hadoop and Java in it.



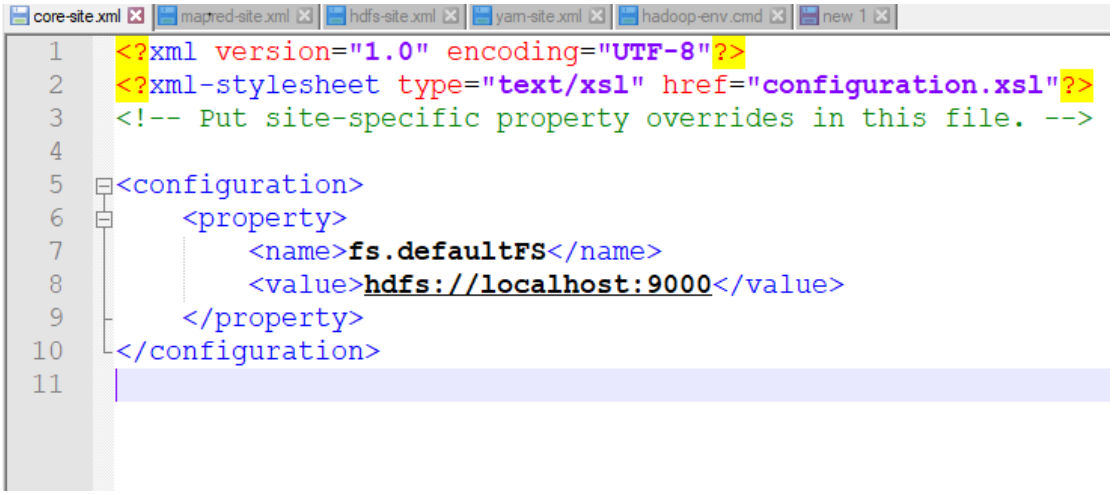
1.3 GUI CONFIGURATION

Now we need to edit some files located in the hadoop directory of the etc folder where we installed hadoop. The files that need to be edited have been highlighted.



1. Edit the file core-site.xml in the hadoop directory. Copy this xml property in the configuration in the file

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



2. Edit mapred-site.xml and copy this property in the configuration

```
<configuration>
        <property>
                <name>mapreduce.framework.name</name>
```

```
            <value>yarn</value>
        </property>
</configuration>
```

[Note : if addition is required, then add the following code

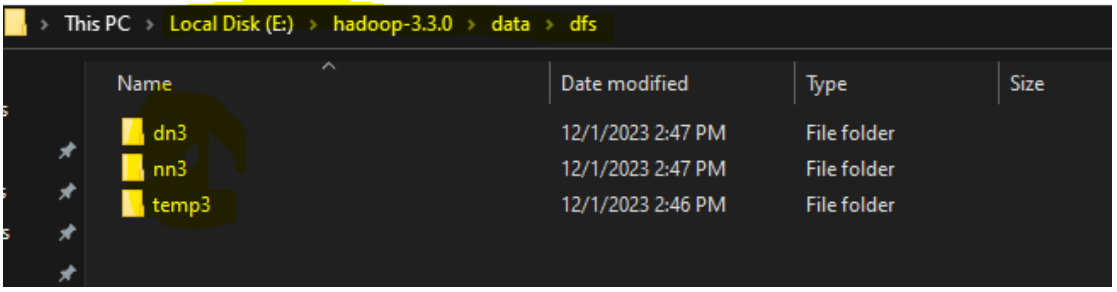
```
        <property>
    <name>mapreduce.application.classpath</name>
<value>%HADOOP_HOME%/share/hadoop/mapreduce/*,%HADOOP_HOME%/share/hadoop/mapreduce/lib/*,%HADOOP_HOME%/share/hadoop/common/*,%HADOOP_HOME%/share/hadoop/common/lib/*,%HADOOP_HOME%/share/hadoop/yarn/*,%HADOOP_HOME%/share/hadoop/yarn/lib/*,%HADOOP_HOME%/share/hadoop/hdfs/*,%HADOOP_HOME%/share/hadoop/hdfs/lib/*</value>
    </property>
```

]

```xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.application.classpath</name>
        <value>%HADOOP_HOME%/share/hadoop/mapreduce/*,%HADOOP_HOME%/share/hadoop/mapreduce/lib/*,
        %HADOOP_HOME%/share/hadoop/common/*,%HADOOP_HOME%/share/hadoop/common/lib/*,
        %HADOOP_HOME%/share/hadoop/yarn/*,%HADOOP_HOME%/share/hadoop/yarn/lib/*,
        %HADOOP_HOME%/share/hadoop/hdfs/*,%HADOOP_HOME%/share/hadoop/hdfs/lib/*</value>
    </property>
</configuration>
```

3. Create a folder 'data' in the hadoop directory

4. Create a folder with the name 'datanode' and a folder 'namenode' in this data directory. [You can create your own folders like dn3, nn3 and temp3. If folders are present already, delete them first]



5. Edit the file hdfs-site.xml and add below property in the configuration

[Note: The path of namenode and datanode across value would be the path of the datanode and namenode folders you just created. **]**

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
          <value>file:///E:/hadoop-3.3.0/data/dfs/nn3</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///E:/hadoop-3.3.0/data/dfs/dn3</value>
  </property>

   <property>
    <name>dfs.permissions.enabled</name>
    <value>true</value>
  </property>
</configuration>
```

core-site.xml | mapred-site.xml | hdfs-site.xml | yarn-site.xml | hadoop-env.cmd | new 1

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <!--<property>
    <name>dfs.datanode.failed.volumes.tolerated</name>
    <value>0</value>
</property>-->
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///E:/hadoop-3.3.0/data/dfs/nn3</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///E:/hadoop-3.3.0/data/dfs/dn3</value>
  </property>

   <property>
    <name>dfs.permissions.enabled</name>
    <value>true</value>
  </property>
</configuration>
```

6. Edit the file yarn-site.xml and add below property in the configuration

```xml
<configuration>
   <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
   </property>
         <property>
 <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>

 </property>
 <property>
   <name>yarn.scheduler.capacity.maximum-am-resource-percent</name>
   <value>1</value>
   <description>
```

```
    Maximum percent of resources in the cluster which can be used to run
    application masters i.e. controls number of concurrent running
    applications.
  </description>
 </property>
  <property>
     <name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,H
ADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_H
OME,HADOOP_MAPRED_HOME</value>
  </property>
</configuration>
```

7. Edit hadoop-env.cmd and replace %JAVA_HOME% with the path of the java folder where your jdk 1.8 is installed.

hadoop-env.cmd - Notepad

File Edit Format View Help

```
@rem WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
@rem See the License for the specific language governing permissions and
@rem limitations under the License.

@rem Set Hadoop-specific environment variables here.

@rem The only required environment variable is JAVA_HOME.  All others are
@rem optional.  When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use.  Required.
set JAVA_HOME=C:\PROGRA~1\Java\jdk1.8.0_91

@rem The jsvc implementation to use. Jsvc is required to run secure datanodes.
@rem set JSVC_HOME=%JSVC_HOME%

@rem set HADOOP_CONF_DIR=
```

8. Hadoop needs Windows OS specific files which do not come with default download of hadoop.

Check whether hadoop is successfully installed by running this command on cmd:

hadoop -version

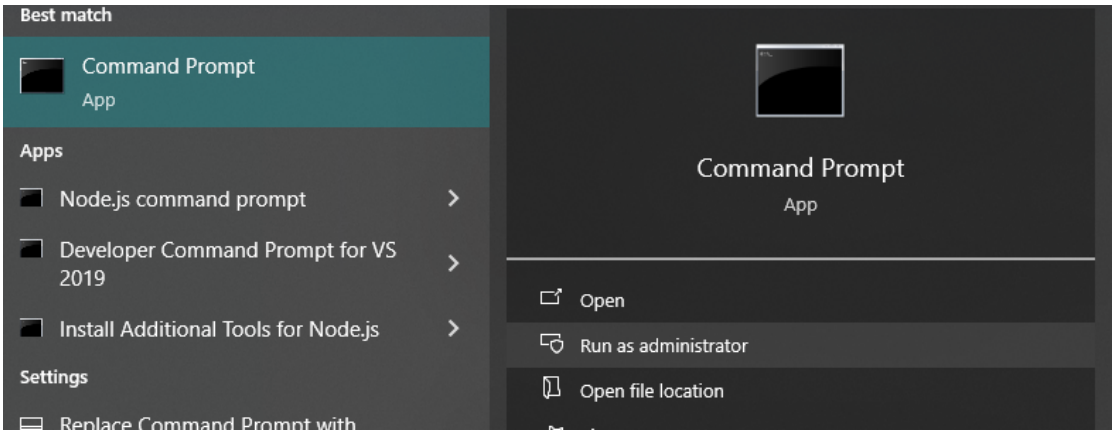
Format the NameNode

Formatting the NameNode is done once when hadoop is installed and not for running hadoop filesystem, else it will delete all the data inside HDFS.

Run this command

hdfs namenode -format

Now change the directory in cmd to sbin folder of hadoop directory with this command, Start namenode and datanode with this command [Run cmd as administrator]:




```
C:\Windows\system32>hdfs namenode -format
2023-12-04 15:59:55,854 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = CC1-04/192.168.203.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = E:\hadoop-3.3.0\etc\hadoop;E:\hadoop-3.3.0\sh
0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;E:\hadoop-
otations-0.5.0.jar;E:\hadoop-3.3.0\share\hadoop\common\lib\avro-1.7.7.ja
mmon\lib\commons-beanutils-1.9.4.jar;E:\hadoop-3.3.0\share\hadoop\common
p-3.3.0\share\hadoop\common\lib\commons-collections-3.2.2.jar;E:\hadoop-
commons-configuration2-2.1.1.jar;E:\hadoop-3.3.0\share\hadoop\common\lib
3.3.0\share\hadoop\common\lib\commons-lang3-3.7.jar;E:\hadoop-3.3.0\shar
th3-3.1.1.jar;E:\hadoop-3.3.0\share\hadoop\common\lib\commons-net-3.6.ja
on\lib\curator-client-4.2.0.jar;E:\hadoop-3.3.0\share\hadoop\common\lib\
\hadoop-3.3.0\share\hadoop\common\lib\dnsjava-2.1.7.jar;E:\hadoop-3.3.0\
```

After some time you will get Datanode or namenode successfully formatted.

```
2023-12-04 15:59:59,775 INFO util.GSet: VM type       = 64-bit
2023-12-04 15:59:59,775 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
2023-12-04 15:59:59,775 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory root= E:\hadoop-3.3.0\data\dfs\nn3; location= null ? (Y or N) y
2023-12-04 16:00:07,433 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1948773041-192.168.203.1-1701685807402
2023-12-04 16:00:07,433 INFO common.Storage: Will remove files: [E:\hadoop-3.3.0\data\dfs\nn3\current\edits_0000000000000000001-0000000
a\dfs\nn3\current\edits_0000000000000000202-0000000000000000367, E:\hadoop-3.3.0\data\dfs\nn3\current\edits_inprogress_0000000000000000
\current\fsimage_0000000000000000201, E:\hadoop-3.3.0\data\dfs\nn3\current\fsimage_0000000000000000201.md5, E:\hadoop-3.3.0\data\dfs\nn
0367, E:\hadoop-3.3.0\data\dfs\nn3\current\fsimage_0000000000000000367.md5, E:\hadoop-3.3.0\data\dfs\nn3\current\seen_txid, E:\hadoop-3
]
2023-12-04 16:00:07,636 INFO common.Storage: Storage directory E:\hadoop-3.3.0\data\dfs\nn3 has been successfully formatted.
2023-12-04 16:00:07,777 INFO namenode.FSImageFormatProtobuf: Saving image file E:\hadoop-3.3.0\data\dfs\nn3\current\fsimage.ckpt_000000
n
2023-12-04 16:00:07,980 INFO namenode.FSImageFormatProtobuf: Image file E:\hadoop-3.3.0\data\dfs\nn3\current\fsimage.ckpt_0000000000000
 0 seconds .
2023-12-04 16:00:08,027 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-12-04 16:00:08,042 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-12-04 16:00:08,058 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at CC1-04/192.168.203.1
************************************************************/

C:\Windows\system32>
```

start-dfs.cmd

Two more cmd windows will open for NameNode and DataNode

Now start yarn through this command

start-yarn.cmd

Note: Make sure all the 4 Apache Hadoop Distribution windows are up n running. If they are not running, you will see an error or a shutdown message. In that case, you need to debug the error.

or just run

start-all.cmd

[It will launch 4 windows of 4 processes namely : Namenode, Datanode, Resource Manager and Data Manager. The cursor should be remain blinking or process stays in running state]

```
2023-12-04 16:08:11,217 INFO util.GSet: VM type       = 64-bit
Apache Hadoop Distribution - hadoop  namenode
Apache Hadoop Distribution - hadoop  datanode
Apache Hadoop Distribution - yarn  resourcemanager
Apache Hadoop Distribution - yarn  nodemanager
Dec 04, 2023 4:10:17 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19 02/11/2015 03:25 AM'
Dec 04, 2023 4:10:17 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider with
 the scope "Singleton"
Dec 04, 2023 4:10:17 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Sin
gleton"
Dec 04, 2023 4:10:17 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the s
cope "Singleton"
2023-12-04 16:10:17,720 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@1f0b3cfe{node,/,file:///C:/Users/admi
n/AppData/Local/Temp/jetty-0_0_0_0-8042-_-any-4071096618355146716.dir/webapp/,AVAILABLE}{jar:file:/E:/hadoop-3.3.0/share
/hadoop/yarn/hadoop-yarn-common-3.3.0.jar!/webapps/node}
2023-12-04 16:10:17,729 INFO server.AbstractConnector: Started ServerConnector@7bb3a9fe{HTTP/1.1,[http/1.1]}{0.0.0.0:804
2}
2023-12-04 16:10:17,730 INFO server.Server: Started @20572ms
2023-12-04 16:10:17,732 INFO webapp.WebApps: Web app node started at 8042
2023-12-04 16:10:17,733 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : CC1-04:62760
2023-12-04 16:10:17,736 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2023-12-04 16:10:17,745 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031
2023-12-04 16:10:17,781 INFO nodemanager.NodeStatusUpdaterImpl: Sending out 0 NM container statuses: []
2023-12-04 16:10:17,875 INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM using containers :[]
2023-12-04 16:10:18,135 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key wi
th id 1621074528
2023-12-04 16:10:18,138 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id
 96496502
2023-12-04 16:10:18,138 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as CC1-04:62760 with tot
al resource of <memory:8192, vCores:8>
starting yarn daemons
```

```
C:\Windows\system32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\system32>
```

To check whether these 4 process are running, we can use jps command.

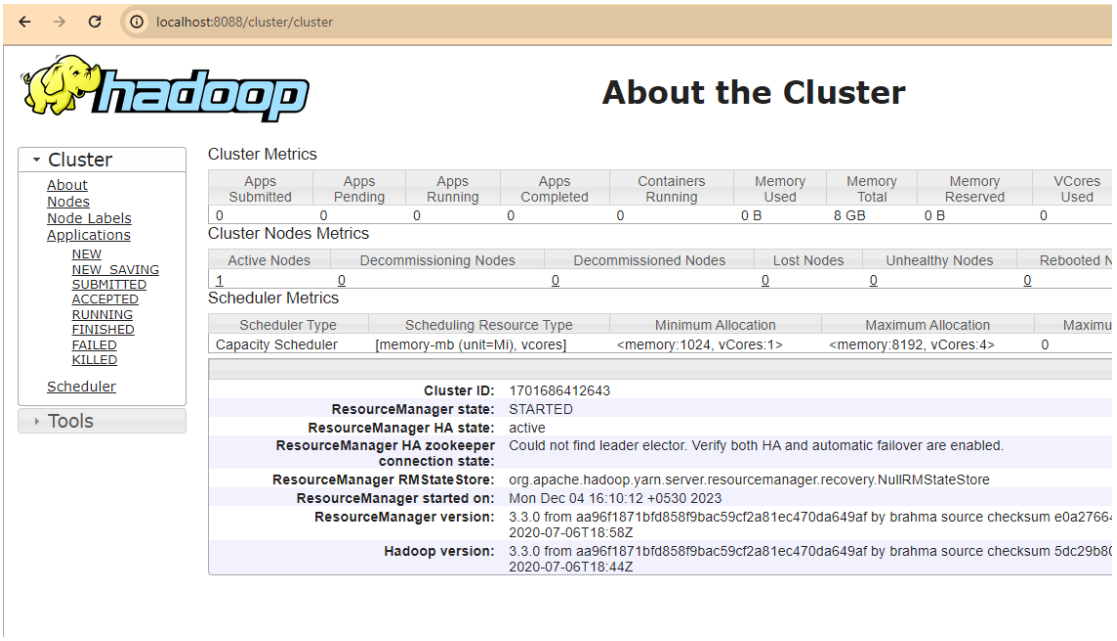
jps

```
C:\Windows\system32>jps
1744 NameNode
5840 ResourceManager
6864 NodeManager
9716 Jps
7208 DataNode
```

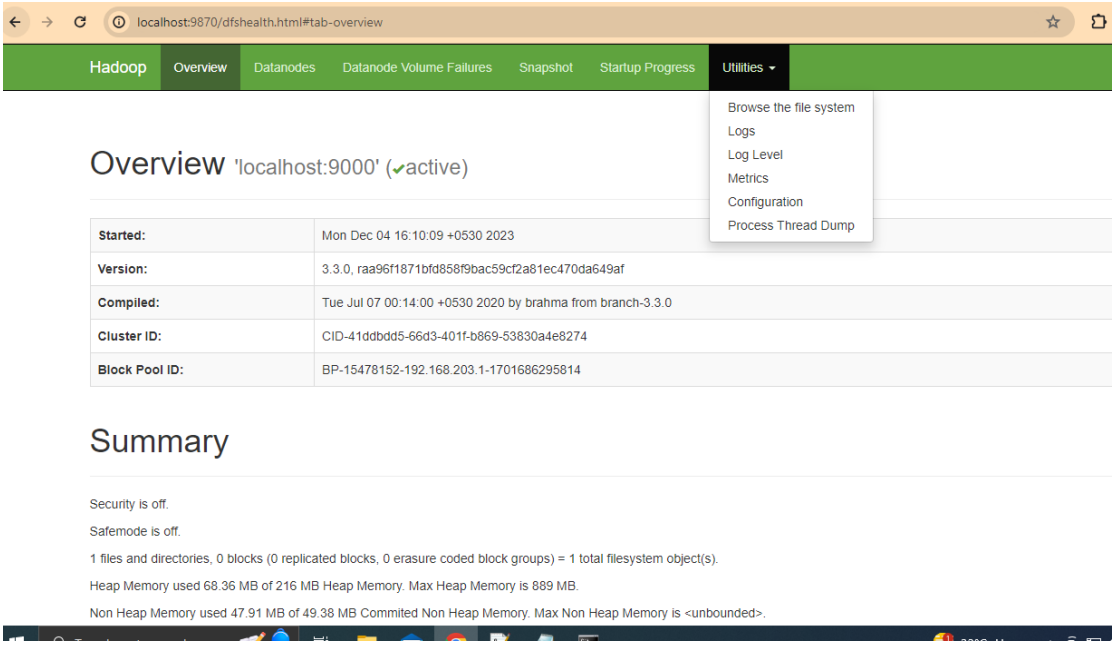
To access information about resource manager current jobs, successful and failed jobs, go to this link in browser

http://localhost:8088/cluster

To check the details about the hdfs (namenode and datanode),



http://localhost:9870/



1.4 COMMAND LINE CONFIGURATION

Hadoop HDFS Commands

With the help of the HDFS commands, 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.

Before starting with the HDFS command, we have to start the Hadoop services. In this practical, we have mentioned the Hadoop HDFS commands with their usage, examples, and description.

1. version

Hadoop HDFS version Command Usage:

hadoop -version

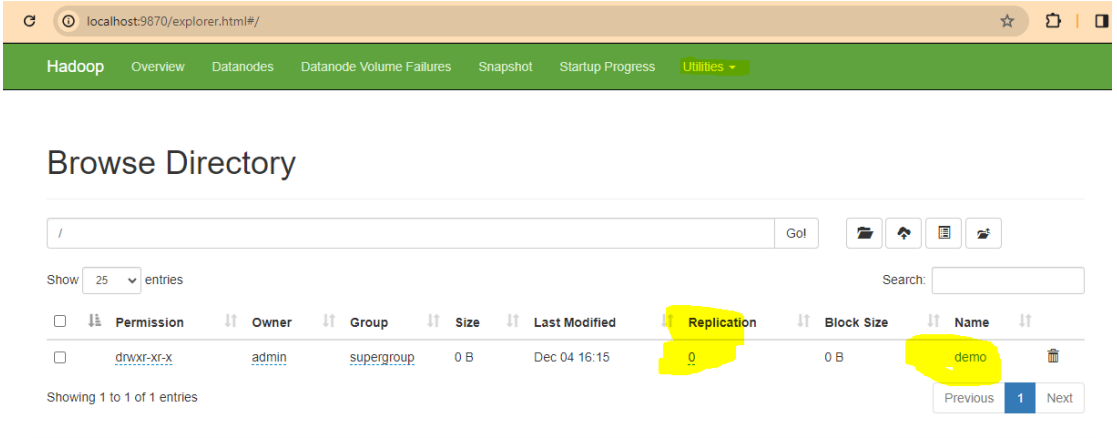
2. mkdir

Hadoop HDFS mkdir Command Usage: *hadoop dfs –mkdir /path/directory_name*

we create a new directory named directory_name in HDFS using the mkdir command.

or use *hdfs dfs –mkdir /path/directory_name*

```
C:\Windows\system32>hadoop dfs -mkdir /demo
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```



3. ls

Hadoop HDFS ls Command Usage: *hadoop dfs -ls /path*

or

hdfs dfs -ls /path

Hadoop HDFS ls Command Description:

The Hadoop fs shell command ls displays a list of the contents of a directory specified in the path provided by the user. It shows the name, permissions, owner, size, and modification date for each file or directories in the specified directory.

4. put

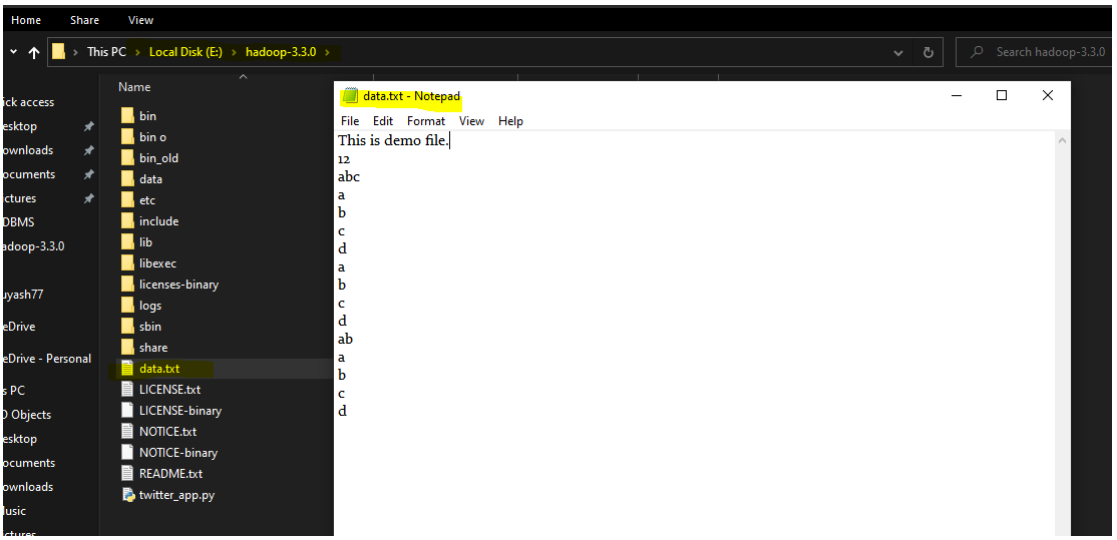
Hadoop HDFS put Command Usage:

haoop dfs -put <localsrc> <dest>

hdfs dfs -put <localsrc> <dest>

Hadoop HDFS put Command Example:

Here in this example, we are trying to copy localfile1 of the local file system to the Hadoop filesystem.

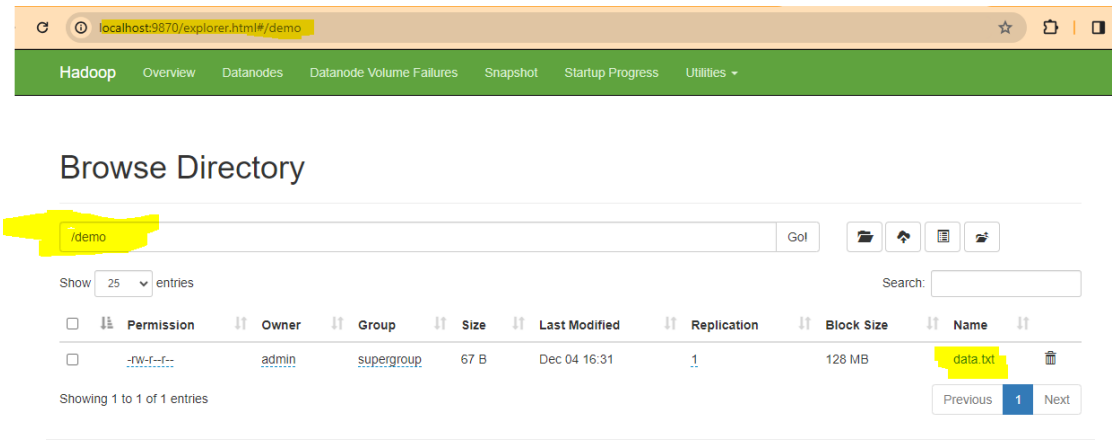



```
C:\Windows\system32>hdfs dfs -put "E:\hadoop-3.3.0\data.txt" /demo

C:\Windows\system32>
```

hdfs dfs -put "E:\hadoop-3.3.0\data.txt" /demo

output will be visible on http://localhost:9870/ **, click on Utilities - > Browse the file system**



5. copyFromLocal

Hadoop HDFS copyFromLocal Command Usage:

hadoop dfs -copyFromLocal <localsrc> <hdfs destination>

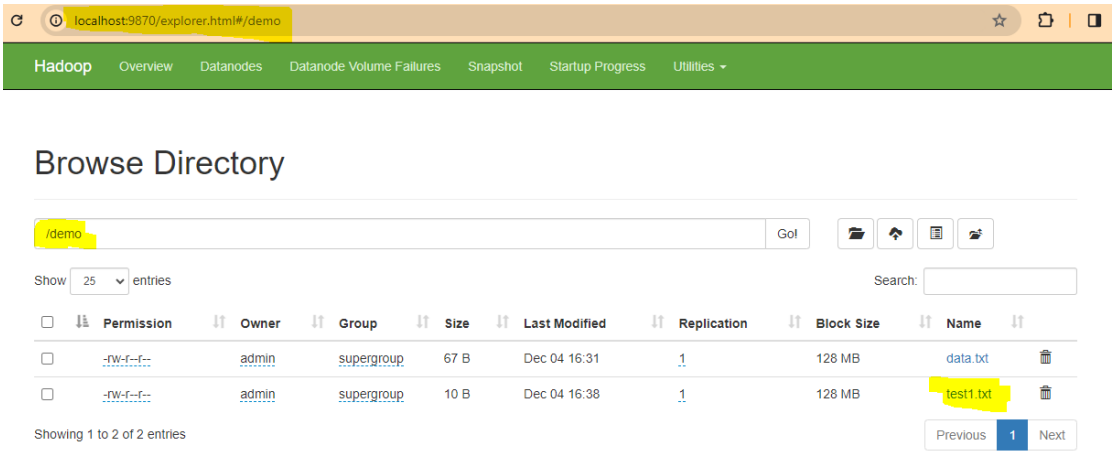
hdfs dfs -copyFromLocal <localsrc> <hdfs destination>

Hadoop HDFS copyFromLocal Command Example:

Here in the below example, we are trying to copy the 'test1' file present in the local file system to the demo directory of Hadoop.

```
C:\Windows\system32>hdfs dfs -copyFromLocal "E:\hadoop-3.3.0\test1.txt" /demo

C:\Windows\system32>
```



6. get

Hadoop HDFS get Command Usage:

hadoop dfs -get <src> <localdest>

hdfs dfs -get <src> <localdest>

Hadoop HDFS get Command Example:

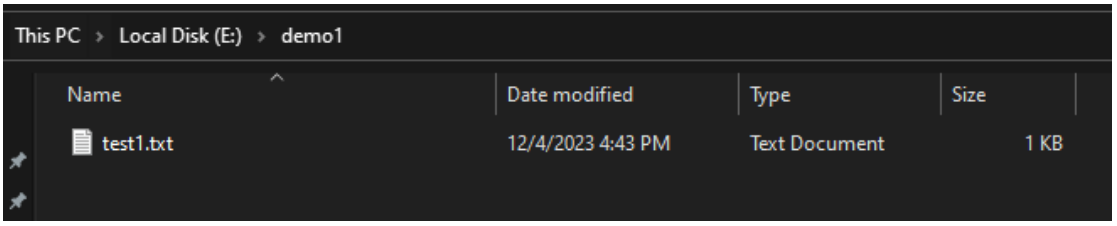
In this example, we are trying to copy the 'test1.txt' of the hadoop filesystem to the local file system.

Hadoop HDFS get Command Description:

The Hadoop fs shell command get copies the file or directory from the Hadoop file system to the local file system.

```
C:\Windows\system32>hdfs dfs -get /demo/test1.txt "E:\demo1"

C:\Windows\system32>
```



7. copyToLocal

Hadoop HDFS copyToLocal Command Usage:

hadoop dfs -copyToLocal <hdfs source> <localdst>

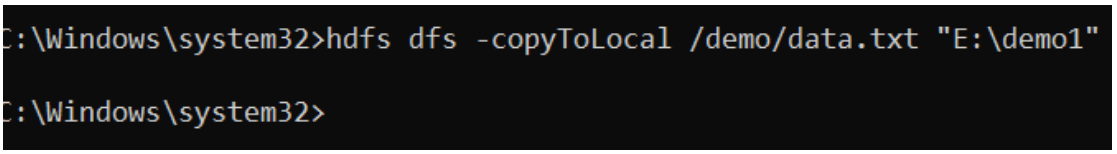
hdfs dfs -copyToLocal <hdfs source> <localdst>

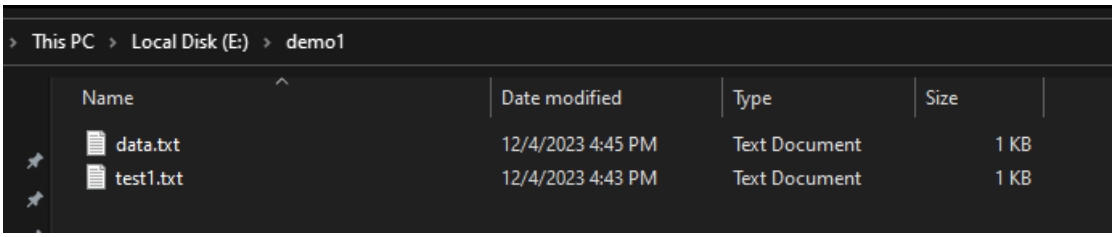
Hadoop HDFS copyToLocal Command Example:

Here in this example, we are trying to copy the 'data.txt' file present in the demo directory of HDFS to the local file system.

hadoop HDFS copyToLocal Description:

copyToLocal command copies the file from HDFS to the local file system.





8. cat

Hadoop HDFS cat Command Usage:

Hadoop dfs –cat /path_to_file_in_hdfs

hdfs dfs –cat /path_to_file_in_hdfs

Hadoop HDFS cat Command Example:

Here in this example, we are using the cat command to display the content of the 'sample' file present in newDataFlair directory of HDFS.

Hadoop HDFS cat Command Description:

The cat command reads the file in HDFS and displays the content of the file on console or stdout.

```
C:\Windows\system32>hdfs dfs -cat /demo/data.txt
This is demo file.
12
abc
a
b
c
d
a
b
c
d
ab
```

9. mv

Hadoop HDFS mv Command Usage:

hadoop dfs -mv <src> <dest>

hdfs dfs -mv <src> <dest>

Hadoop HDFS mv Command Example:

In this example, we have a directory 'demo' in HDFS. We are using mv command to move the demo directory to the BigDemo directory in HDFS.

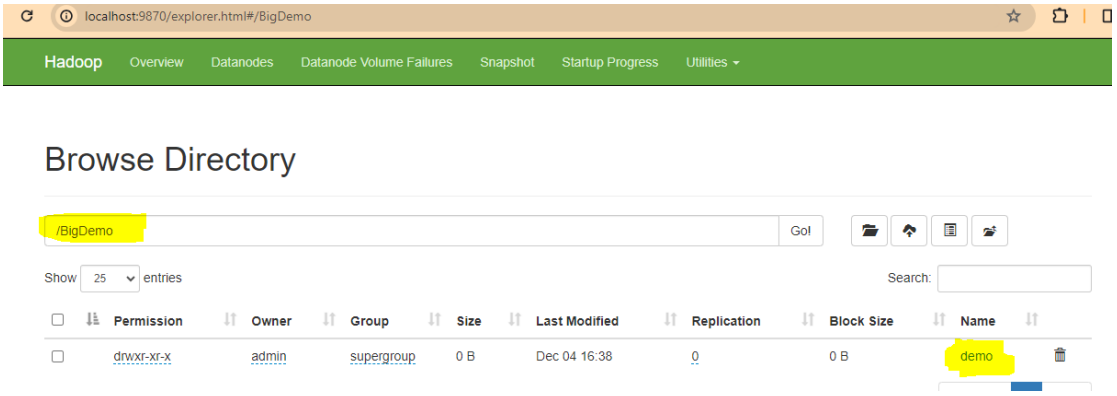
Hadoop HDFS mv Command Description:

The HDFS mv command moves the files or directories from the source to a destination within HDFS.

```
C:\Windows\system32>hdfs dfs -mkdir /BigDemo

C:\Windows\system32>hdfs dfs -mv /demo /BigDemo

C:\Windows\system32>
```



10. cp

Hadoop HDFS cp Command Usage:

hadoop dfs -cp <src> <dest>

hdfs dfs -cp <src> <dest>

Hadoop HDFS cp Command Example:

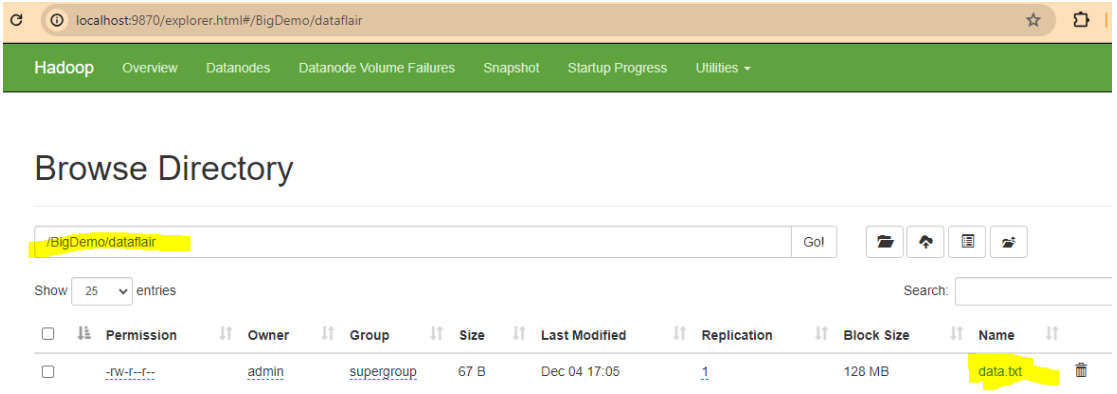
In the below example we are copying the 'file1' present in demo directory in HDFS to the dataflair directory of HDFS.

Hadoop HDFS cp Command Description:

The cp command copies a file from one directory to another directory within the HDFS.

```
C:\Windows\system32>hdfs dfs -cp /BigDemo/demo/data.txt /BigDemo/dataflair

C:\Windows\system32>
```



1.5 SUMMARY

With this practical, we are now able to:

1. Install hadoop on windows
2. run several commands of hadoop

1.6 REFERENCES

1. https://kontext.tech/article/447/install-hadoop-330-on-windows-10-step-by-step-guide
2. https://projectsbasedlearning.com/bigdata-hadoop/apache-hadoop-3-3-0-single-node-installation-on-windows-10-part-2/ [preferred]