

## 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
- 1.2 Prerequisite
- 1.3 GUI Configuration
- 1.4 Command Line Configuration
- 1.5 Summary
- 1.6 Sample Questions
- 1.7 References

---

#### 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

---

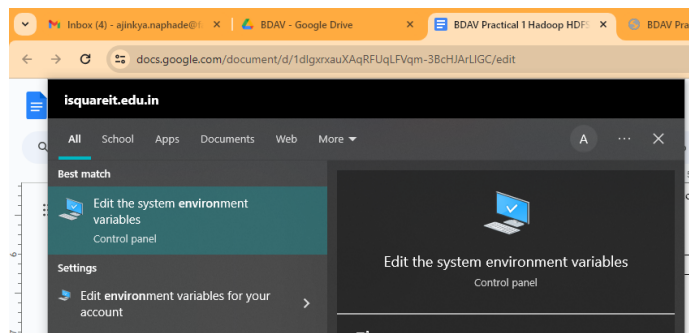
#### 1.2 PREREQUISITE

---

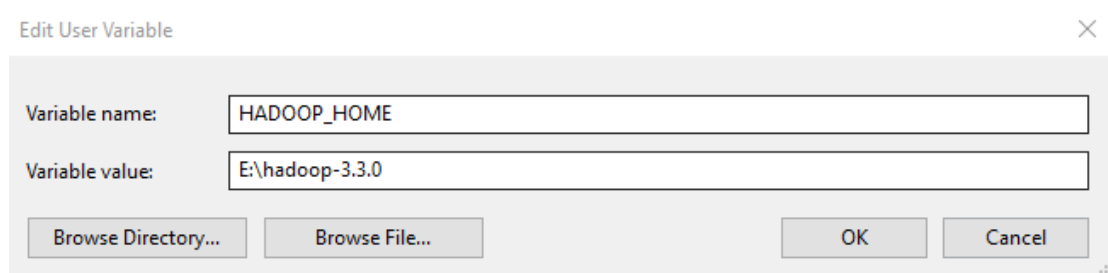
Check your java version through this command on command prompt.

##### **java -version**

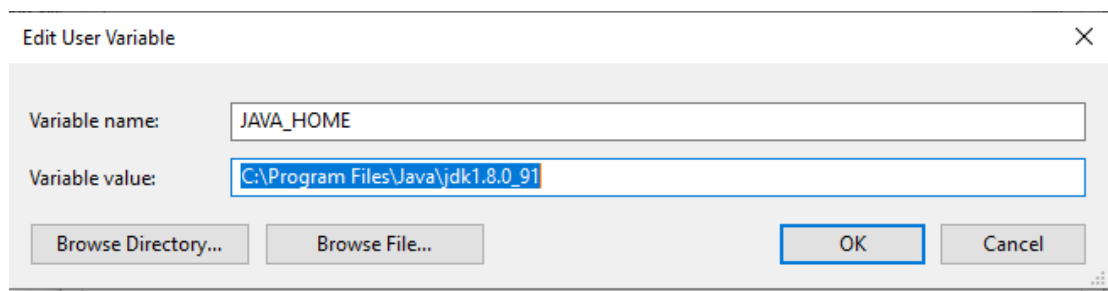
Create a new user variable. Put the Variable\_name as HADOOP\_HOME and Variable\_value as the path of the bin folder where you extracted hadoop.



Enter administrative details as per need.

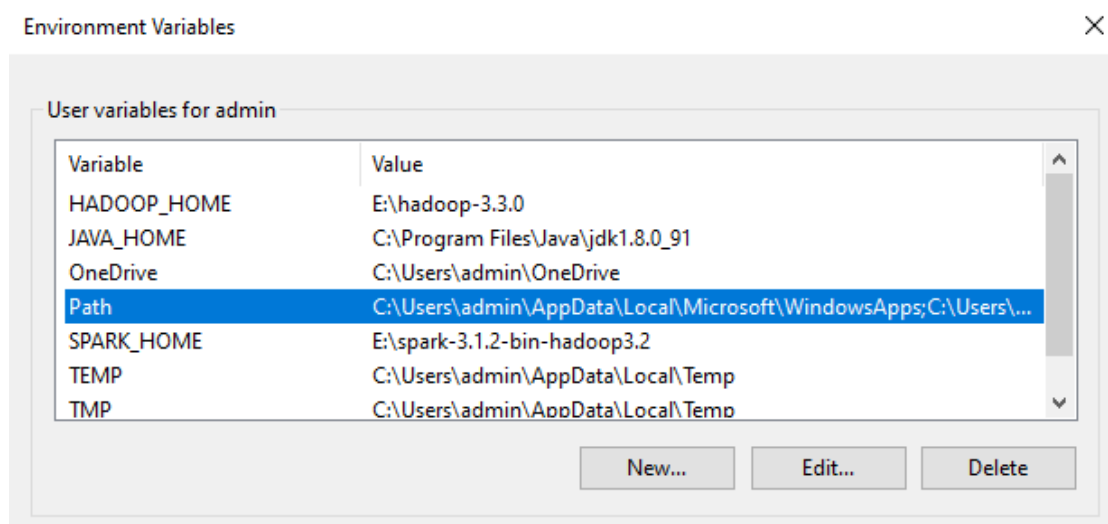


Likewise, create a new user variable with variable name as JAVA\_HOME and variable value as the path of the bin folder in the Java directory.

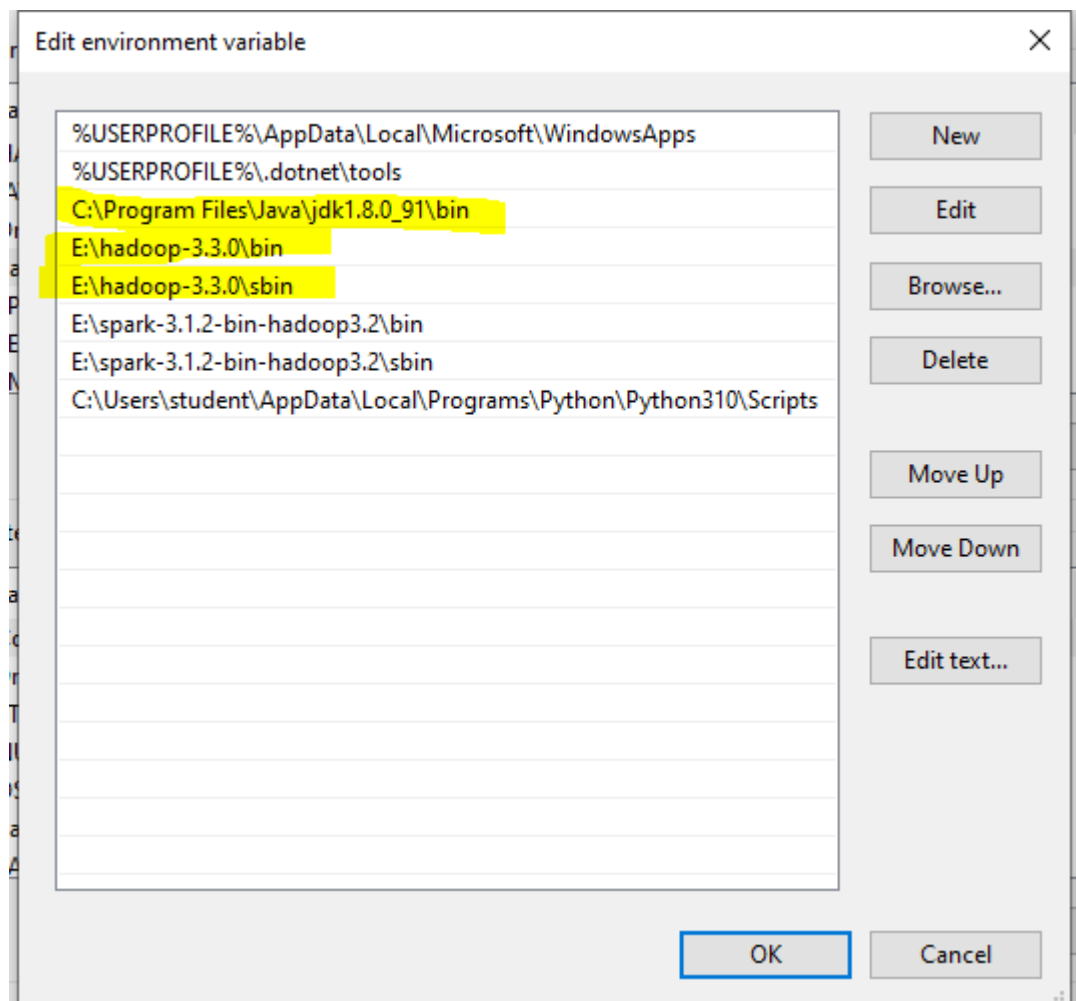


Now we need to set Hadoop bin directory and Java bin directory path in system variable path.

Edit Path in system variable :



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

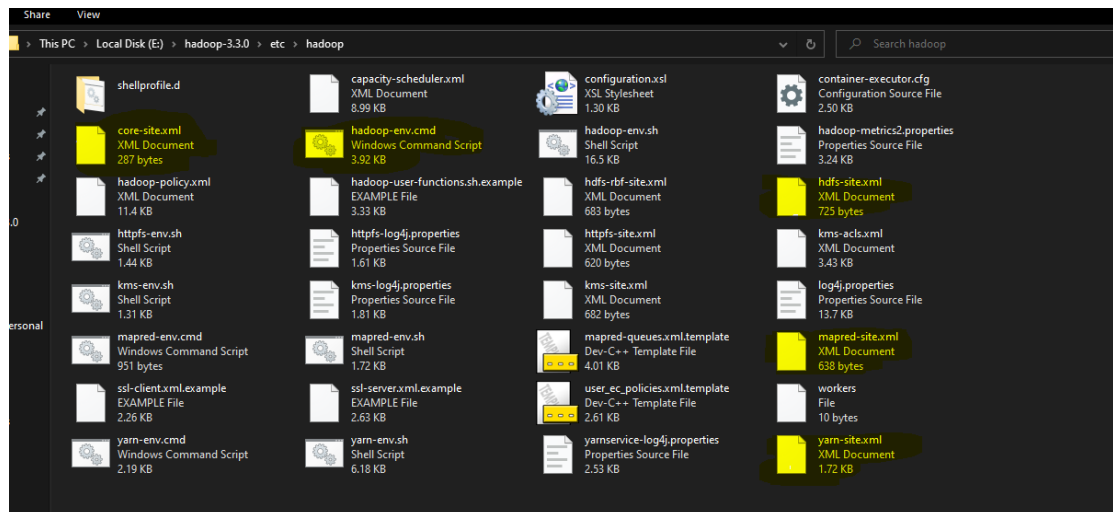


---

### 1.3 GUI CONFIGURATION

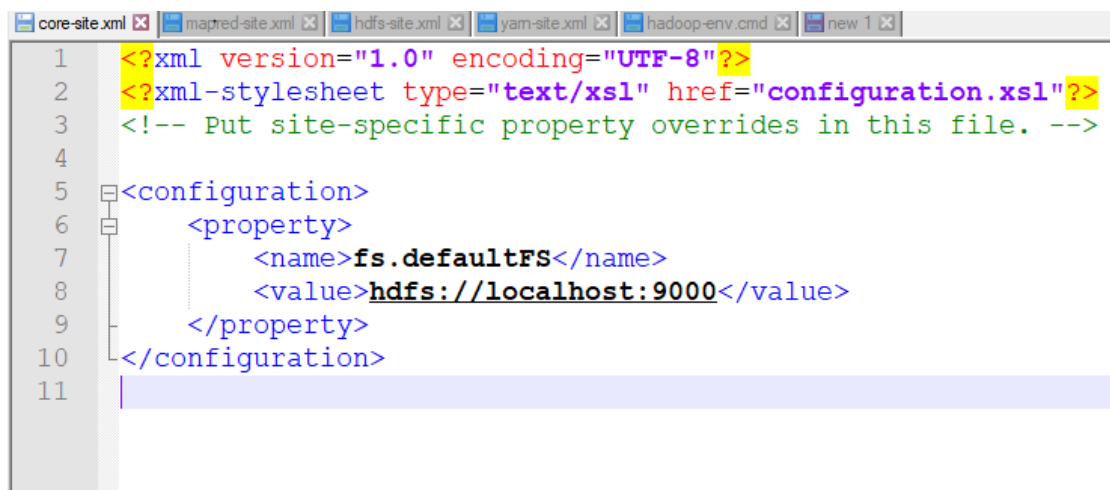
---

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>

```

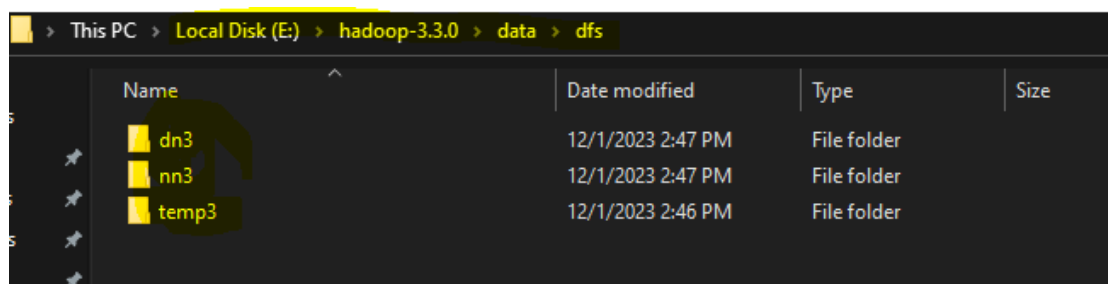
]

```

1  <?xml version="1.0"?>
2  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3  <configuration>
4      <property>
5          <name>mapreduce.framework.name</name>
6          <value>yarn</value>
7      </property>
8      <property>
9          <name>mapreduce.application.classpath</name>
10         <value>%HADOOP_HOME%/share/hadoop/mapreduce/*,%HADOOP_HOME%/share/hadoop/mapreduce/lib/*,
11             %HADOOP_HOME%/share/hadoop/common/*,%HADOOP_HOME%/share/hadoop/common/lib/*,
12             %HADOOP_HOME%/share/hadoop/yarn/*,%HADOOP_HOME%/share/hadoop/yarn/lib/*,
13             %HADOOP_HOME%/share/hadoop/hdfs/*,%HADOOP_HOME%/share/hadoop/hdfs/lib/*</value>
14     </property>
15 </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>
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4  <!-- Put site-specific property overrides in this file. -->
5  <configuration>
6    <property>
7      <name>dfs.replication</name>
8      <value>1</value>
9    </property>
10   <!--<property>
11     <name>dfs.datanode.failed.volumes.tolerated</name>
12     <value>0</value>
13   </property-->
14   <property>
15     <name>dfs.namenode.name.dir</name>
16     <value>file:///E:/hadoop-3.3.0/data/dfs/nn3</value>
17   </property>
18   <property>
19     <name>dfs.datanode.data.dir</name>
20     <value>file:///E:/hadoop-3.3.0/data/dfs/dn3</value>
21   </property>
22
23   <property>
24     <name>dfs.permissions.enabled</name>
25     <value>true</value>
26   </property>
27 </configuration>

```

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

```

<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>

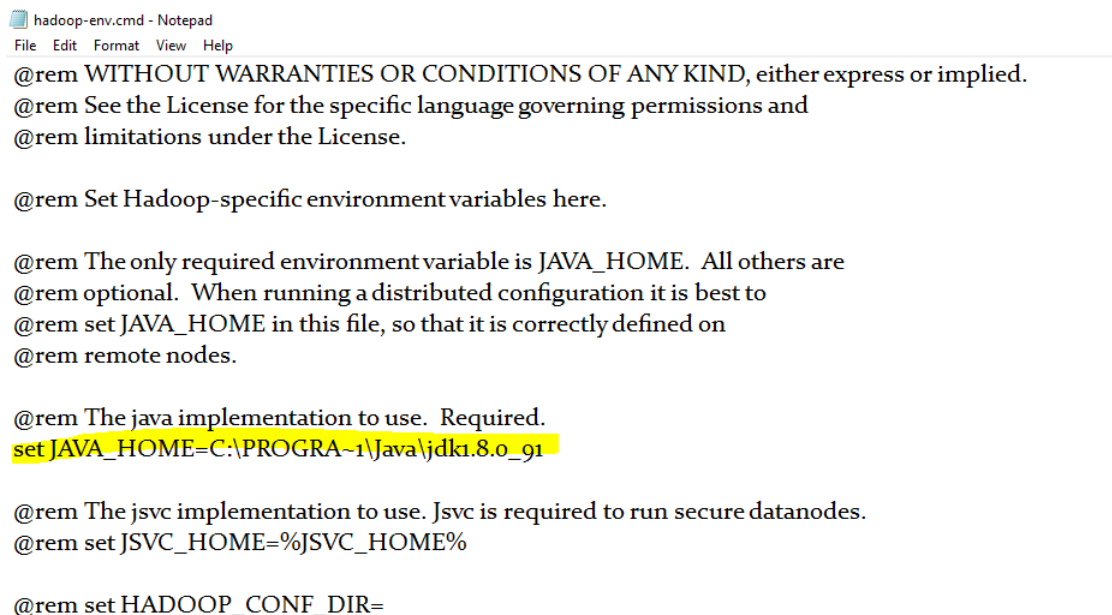
  <property>
    <name>yarn.scheduler.capacity.maximum-am-resource-percent</name>
    <value>1</value>
    <description>

```

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:\PROGRAMS\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**

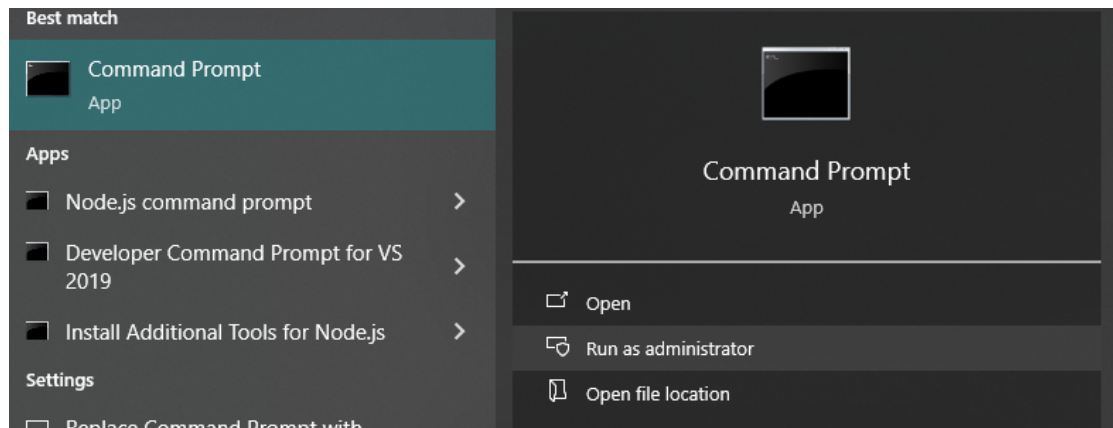


Formatting the NameNode is done once when hadoop is installed and not for running hadoop filesystem, else it will delete all the data inside HDFS.

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\
ian;E:\hadoop-3.3.0\share\hadoop\common\lib\guava-27.0-jre.jar;E:\hadoo
```

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



```
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.

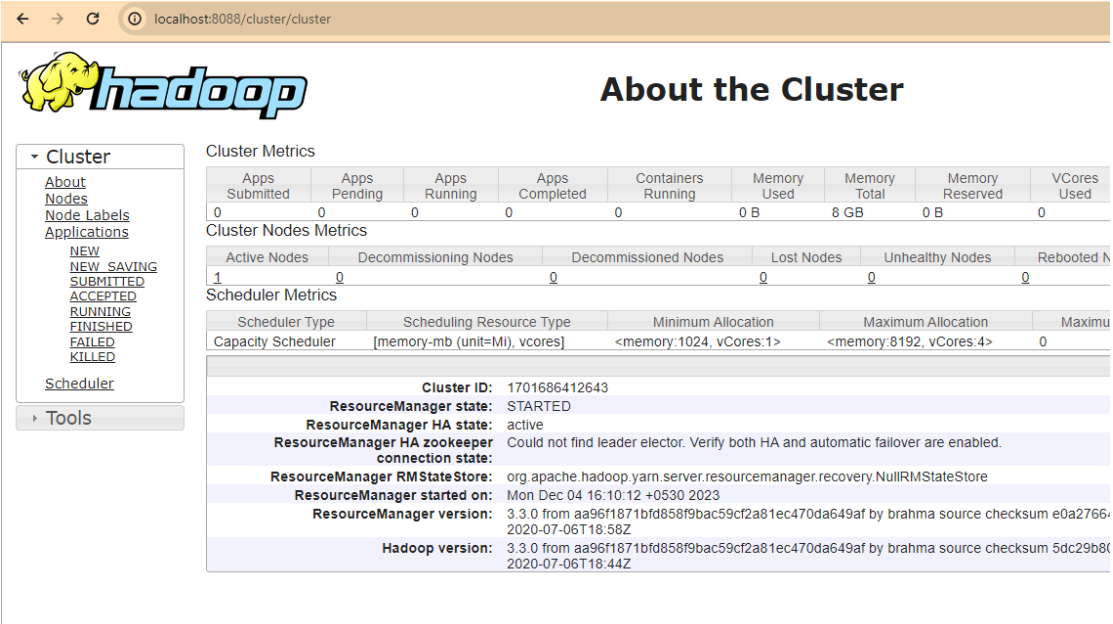
jps

```
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),



The screenshot displays the Hadoop cluster management web interface. The left sidebar contains a navigation menu with options like Cluster, About, Nodes, Node Labels, Applications, and Tools. The main content area is titled 'About the Cluster' and includes several tables and configuration details.

**Cluster Metrics**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used
0	0	0	0	0	0 B	8 GB	0 B	0

**Cluster Nodes Metrics**

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted N
1	0	0	0	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximu
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

**Cluster Configuration Details:**

- Cluster ID: 1701686412643
- ResourceManager state: STARTED
- ResourceManager HA state: active
- ResourceManager HA zookeeper connection state: Could not find leader elector. Verify both HA and automatic failover are enabled.
- ResourceManager RMStateStore: org.apache.hadoop.yarn.server.resourcemanager.recovery.NullRMStateStore
- ResourceManager started on: Mon Dec 04 16:10:12 +0530 2023
- ResourceManager version: 3.3.0 from aa96f1871bfd858f9bac59cf2a81ec470da649af by brahma source checksum e0a2766-2020-07-06T18:58Z
- Hadoop version: 3.3.0 from aa96f1871bfd858f9bac59cf2a81ec470da649af by brahma source checksum 5dc29b8f-2020-07-06T18:44Z

http://localhost:9870/

Overview 'localhost:9000' (✓active)

Started:	Mon Dec 04 16:10:09 +0530 2023
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0
Cluster ID:	CID-41ddb5-66d3-401f-b869-53830a4e8274
Block Pool ID:	BP-15478152-192.168.203.1-1701686295814

Summary

Security is off.  
Safemode is off.  
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).  
Heap Memory used 68.36 MB of 216 MB Heap Memory. Max Heap Memory is 889 MB.  
Non Heap Memory used 47.91 MB of 49.38 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

## 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.
```

The screenshot shows the Hadoop web interface at localhost:9870/explorer.html. The 'Browse Directory' section displays a table of files in the root directory. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. A single entry is shown: a directory named 'demo' with permissions 'drwxr-xr-x', owner 'admin', group 'supergroup', size '0 B', last modified 'Dec 04 16:15', and a replication factor of '0'. The 'demo' directory name and its replication factor are highlighted in yellow. The interface also includes a search bar, a 'Go!' button, and pagination controls showing '1' of 1 entries.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	admin	supergroup	0 B	Dec 04 16:15	0	0 B	demo

### 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

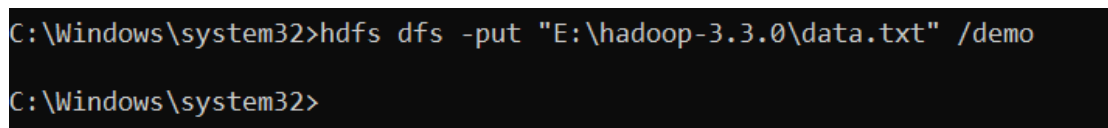
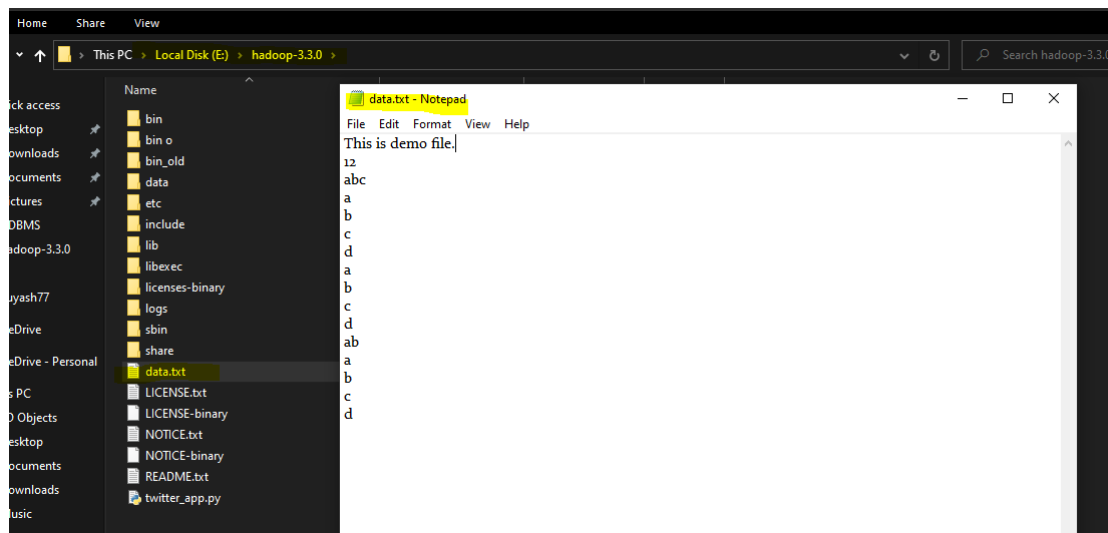
Hadoop HDFS put Command Usage:

`hadoop dfs -put <localsrc> <dest>`

`hdfs dfs -put <localsrc> <dest>`

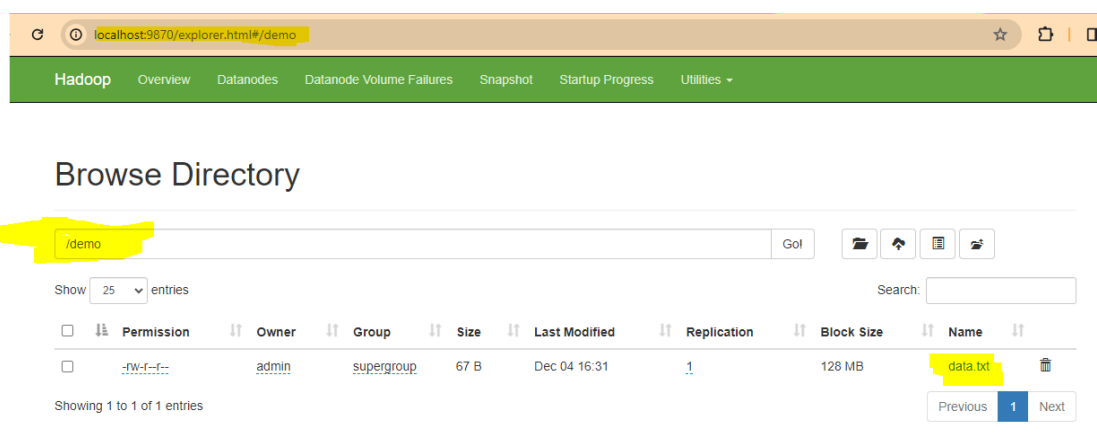
Hadoop HDFS put Command Example:

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



`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>
```

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-f--	admin	supergroup	67 B	Dec 04 16:31	1	128 MB	data.txt
-rw-f--	admin	supergroup	10 B	Dec 04 16:38	1	128 MB	test1.txt

## 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>
```

Name	Date modified	Type	Size
test1.txt	12/4/2023 4:43 PM	Text Document	1 KB

## 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.

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

C:\Windows\system32>
```

Name	Date modified	Type	Size
data.txt	12/4/2023 4:45 PM	Text Document	1 KB
test1.txt	12/4/2023 4:43 PM	Text Document	1 KB

## 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>
```

The screenshot shows the Hadoop web interface at localhost:9870/explorer.html#/BigDemo. The 'Browse Directory' section displays a table of files and directories. The 'demo' directory is highlighted in yellow.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	admin	supergroup	0 B	Dec 04 16:38	0	0 B	demo

## 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>
```

The screenshot shows the Hadoop web interface at localhost:9870. The breadcrumb path is '/BigDemo/dataflair'. Below the path, there is a table of files. The table has columns: Permission, Owner, Group, Size, Last Modified, Replication, Block Size, Name, and an icon column. The file 'data.txt' is listed with a size of 67 B, last modified on Dec 04 17:05, and 1 replication. The file name 'data.txt' is highlighted in yellow.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	Icon
-rw-r--r--	admin	supergroup	67 B	Dec 04 17:05	1	128 MB	data.txt	

---

## 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 ]