

DSBDA HDFS

1. Check if hadoop is installed or not - hadoop version
2. If yes then check all the nodes are running are no - jps

Format the name node

- cd hadoop-2.7.3
- bin/hadoop namenode -format

- cd hadoop-2.7.3/sbin

Name node

- ./hadoop-daemon.sh start namenode

Data node

- ./hadoop-daemon.sh start datanode

Resource manager-

- ./yarn-daemon.sh start resourcemanager

Node manager-

- ./yarn-daemon.sh start nodemanager

History server-

- ./mr-jobhistory-daemon.sh start historyserver

3. If no install hadoop and then start all the nodes

Step 1) Install java

(sudo apt-get install openjdk-8-jdk)

Step 2) Download the Hadoop 2.7.3 Package.

wget <https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

Step 3) Extract the Hadoop tar File.

tar -xvf hadoop-2.7.3.tar.gz

Step 4) Add the Hadoop and Java paths in the bash file (.bashrc).

gedit .bashrc

export HADOOP_HOME=\$HOME/hadoop-2.7.3

```
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3
export YARN_HOME=$HOME/hadoop-2.7.3
export PATH=$PATH:$HOME/hadoop-2.7.3/bin
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=/usr/lib/jvm/java-8-openjdk-amd64/bin:$PATH
```

Step 5) For applying all these changes to the current Terminal, execute the source command.

```
source .bashrc
```

Step 6) hadoop version

Step 7) Edit the Hadoop Configuration files.

```
cd hadoop-2.7.3/etc/hadoop/
```

Step 8) Open core-site.xml and edit the property mentioned below inside configuration tag:

```
gedit core-site.xml
```

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

Step 9) Edit hdfs-site.xml and edit the property mentioned below inside configuration tag:

```
gedit hdfs-site.xml
```

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>>false</value>
</property>
</configuration>
```

Step 10) Edit the mapred-site.xml file and edit the property mentioned below inside configuration

tag:

```
gedit mapred-site.xml
```

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

Step 11) Edit yarn-site.xml and edit the property mentioned below inside configuration tag:

```
gedit yarn-site.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>
</configuration>
```

Step 12) Edit hadoop-env.sh and add the Java Path as mentioned below:

```
gedit hadoop-env.sh
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Step 13) Start all the nodes by following step 2

- 4. Now create a folder named WordCount and implement your java code there**
- 5. Download the hadoop jar file and put in that folder**

<http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1>

Or u can copy it from the hadoop folder in the home directory in ur folder

- 6. Then create a input folder in your WordCount folder and put input.txt in that folder with some words**
- 7. javac -classpath /home/<ur username>/<folder name here Word Count>/hadoop-core-1.2.1.jar
/home/<Username>/<Foldername>/WordCount.java**
- 8. This will compile ur java code and form the classes**
- 9. jar -cvf words.jar -c WordCount/ .**
- 10. This will create words.jar file in your folder and put all the classes in that**
- 11. hadoop fs -mkdir /input**
- 12. hadoop fs -put WordCount/input/input.txt /input**
- 13. hadoop fs -ls /input**
- 14. hadoop jar /home/<username>/<foldername>/<words>.jar
WordCount /input/input1.txt /out321**
- 15. hadoop fs -ls /out321**
- 16. hadoop fs -cat /out321/part-r-00000**
- 17. (Otherwise check in Browsing HDFS -> Utilities -> Browse the file System -> /)**