Setting up a Single Node Hadoop Cluster

A single node cluster means only one DataNode running and setting up all the NameNode, DataNode, ResourceManager, and NodeManager on a single machine. This is used for studying and testing purposes.

While in a **Multi-node cluster**, there are more than one DataNode running and each DataNode is running on different machines. The multi-node cluster is practically used in organizations for analyzing Big Data. Considering the above example, in real-time when we deal with petabytes of data, it needs to be distributed across hundreds of machines to be processed. Thus, here we use a multi-node cluster.

Hadoop Installation:

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) Go to Hadoop home directory and format the NameNode.

```
cd hadoop-2.7.3
```

```
bin/hadoop namenode -format
```

Step 14) Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons/nodes.

```
cd hadoop-2.7.3/sbin
```

1) Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

```
./hadoop-daemon.sh start namenode
```

2) Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

```
./hadoop-daemon.sh start datanode
```

3) Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

```
./yarn-daemon.sh start resourcemanager
```

4) Start NodeManager:

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

```
./yarn-daemon.sh start nodemanager
```

5) Start JobHistoryServer:

JobHistoryServer is responsible for servicing all job history related requests from client.

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

Step 15) To check that all the Hadoop services are up and running, run the below command.

```
jps
```

Step 16) Now open the Mozilla browser and go to **localhost:50070/dfshealth.html** to check the NameNode interface.

You have successfully installed a single-node Hadoop cluster.