

Multi Node Cluster Setup on Ubuntu

This document explain required steps for setting up a distributed, *multi-node* Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux.

We will start with two node cluster. You can add as many nodes as you want. Let's go into details of each step.

Here are the steps needed:

1. Setup two single node cluster
2. Setup Networking
3. SSH access setup
4. Configuration
 - a) Masters
 - b) Slaves
 - c) Configure conf/core-site.xml
 - d) Configure conf/mapred-site.xml
 - e) Configure conf/hdfs-site.xml
5. Starting Multi-node cluster
 - a) HDFS Daemons
 - b) MapReduce Daemons
6. Stopping multinode cluster
 - a) MapReduce Daemons
 - b) HDFS Daemons
7. Running a MapReduce Job

Step 1 - Setup two single node cluster:

As explained in "single node cluster setup document" start with configuring two single node cluster.

Stop all services on both of the nodes using `bin/stop-all.sh`.

We will setup NameNode and JobTracker on one of the node and will call that master node.

We will setup DataNode and TaskTracker on both of the node and will call the second node as slave node which does not have any of the master.

Step 2 - Setup Networking :

All nodes in cluster should be able to communicate with each other. For simplicity we can keep both of nodes in the same network.

Assign IP `192.168.0.1` to the master and `192.168.0.2` to the slave machine. For all practical purposes we will call node which have master node installed as master and node which is slave-only node as slave.

Add below lines in `/etc/hosts` files to identify master and slave.

```
192.168.0.1    master
```

```
192.168.0.2    slave
```

Step 3 - Password less SSH setup:

Master node need to connect with all slave through password less SSH hence we need to setup password less access.

For this we need to copy the public key generated on master to `authorized_keys` on each slave.

On the master machine execute below command for each slave node. This command will ask for password for `hduser` on slave node.

```
hduser@master:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave
```

Step 4 - Configuration

1. Masters: We need to configure master details in conf/masters file on master node. This file should contain IP of master node or hostname. Add below line in conf/masters file.

```
master
```

2. Slaves: We need to configure slave details in conf/slaves file on master node. This file should contain IP of all slave nodes or hostname. In our case both master and slave node is hosting datanode and tasktracker so we will add both of them in conf/slaves. Any additional slaves can also be added in this file. Add below lines in conf/slaves file.

```
master
```

```
slave
```

```
additioinalSlave1
```

3. Configure conf/core-site.xml: Add following configuration in conf/core-site.xml on all nodes (master and slave) between <configuration></configuration>. This specify namenode host and port.

```
<property>
```

```
    <name>fs.default.name</name>
```

```
    <value>hdfs://master: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>
```

4. Configure conf/mapred-site.xml: Add following configuration in conf/ mapred-site.xml on all nodes (master and slave) between <configuration></configuration>. This specify jobtracker host and port.

```
<property>

    <name>mapred.job.tracker</name>
    <value>master: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>
```

5. Configure conf/hdfs-site.xml: We need to change replication factor to 2 because we have only 2 nodes. Change conf/hdfs-site.xml on master node to change replication factor.

```
<property>

    <name>dfs.replication</name>
    <value>2</value>
    <description>
        Default block replication. The actual number of replications can be specified
        when the file is created. The default is used if replication is not specified in
        create time.
    </description>
</property>
```

Step 5 - Starting Multi-node cluster

1. HDFS Daemons: Run following command only on master node to start HDFS daemon on all machines (Namenode and data node). Check logs in /usr/local/hadoop/logs/ directory.

```
bin/start-dfs.sh
```

2. MapReduce Daemons: Run following command only on master node to start mapreduce daemon on all machines (Namenode and data node).

```
bin/start-mapred.sh
```

Step 6 - Stopping multi-node cluster

1. MapReduce Daemons : Run following command only on master node to stop mapreduce daemon on all machines (Namenode and data node).

```
bin/stop-mapred.sh
```

2. HDFS Daemons: Run following command only on master node to stop HDFS daemon on all machines (Namenode and data node).

```
bin/stop-mapred.sh
```

Step 7 - Running a MapReduce Job

1. Now running a MapReduce job is very simple. Just run the below command. It will start a map reduce job.

```
hduser@master:/usr/local/hadoop$ bin/hadoop jar hadoop*examples*.jar wordcount  
/user/hduser/gutenberg /user/hduser/gutenberg-output
```