

Commissioning and Decommissioning Nodes in a Hadoop Cluster

One of the most attractive features of Hadoop framework is its utilization of commodity hardware. However, this leads to frequent Data Node crashes in a Hadoop cluster. Another striking feature of Hadoop Framework is the ease of scale in accordance to the rapid growth in data volume. Because of these two reasons, one of the most common task of a Hadoop administrator is to commission (Add) and decommission (Remove) Data Nodes in a Hadoop Cluster.

You cannot directly remove any datanode in large cluster or a real-time cluster, as it will cause a lot of disturbance. And if you want to take a machine away for hardware up-gradation purpose, or if you want to bring down one or more than one node, decommissioning will be required because you cannot suddenly shut down the datanode/slave-nodes. Similarly, if you want to scale your cluster or add new data nodes without shutting down the cluster, you need commissioning.

We can decommission the node by following below steps or using ambari UI.

Steps for Decommissioning of datanode.

First make sure that no jobs are running on the datanode which you want to decommission.

You can verify this by accessing resource manager GUI.

Create the file `$HADOOP_HOME/conf/dfs-exclude.txt` with the following content:

```
slave1
```

The `dfs-exclude.txt` file contains the DataNode hostnames, one per line, that are to be decommissioned from the cluster

Add the following property to the file `$HADOOP_HOME/conf/hdfs-site.xml`:

```
<property>
<name>dfs.hosts.exclude</name>
<value>$HADOOP_HOME/conf/dfs-exclude.txt</value>
</property>
```

Force the NameNode to reload the active DataNodes using the following command:

```
hadoop dfsadmin refreshNodes
```

Now the decommission process will be started and we can verify the status of data nodes in namenode webUI.

Get a description report of each active DataNode with the following command

```
hadoop dfsadmin report
```

Steps for Decommissioning of datanode using Ambari UI

Logging to ambari gui and go to the click on tab, then select the host on which the datanode needs to be decommissioned:-

Then click on **Action** button on the upper left side of the screen, then on ***selected hosts, then*** on DataNode then decommission

Actions ▾

Filter: All (4) ▾

Name ▲		IP Address ▴▾	Rack ▴▾
<input type="checkbox"/>	<input type="text" value="Any"/>	<input type="text" value="Any"/>	<input type="text" value="Any"/>
<input type="checkbox"/>	✓ impetus-i0095.impetus.co.in	172.26.60.19	/defau
<input type="checkbox"/>	✓ impetus-i0161.impetus.co.in 1	172.26.60.16	/defau
<input checked="" type="checkbox"/>	✓ impetus-i0163.impetus.co.in	172.26.60.17	/defau
<input type="checkbox"/>	✓ impetus-i0203.impetus.co.in 2	172.26.60.18	/defau
			1

Actions ▾ Filter: All (4) ▾

- + Add New Hosts
- Selected Hosts (1) ▸
- Filtered Hosts (4) ▸
 - Hosts ▸
 - DataNodes ▸
 - NodeManagers ▸
- All Hosts (4) ▸

		IP Address ▴ ▾	Rack ▴ ▾
<input type="checkbox"/>	✓ impetus-i0161.impetus.co.in		Any
<input checked="" type="checkbox"/>	✓ impetus-i0163.impetus.co.in	172.2	Any
<input type="checkbox"/>	✓ impetus-i0203.impetus.co.in 2	172.2	

- Start
- Stop
- Restart
- Decommission
- Recommission

Licensed under the Apache License, Version 2.0.
See [third-party tools/resources that Ambari uses and their respective licenses](#)

One window appears for confirmation, click ok. It will start decommissioning the node

After decommission is done click on ***Host Actions*** in right hand side of the window, and click on the delete host. (if required*)

✓ **impetus-i0163.impetus.co.in**

← Back

Summary

Configs

Alerts 0

Versions

Components

✓ Metrics Collector / [Ambari Metrics](#)

✓ Grafana / [Ambari Metrics](#)

✓ ZooKeeper Server / [ZooKeeper](#)

✗ DataNode / [HDFS](#)

✓ SmartSense HST Agent / [SmartSense](#)

✓ JournalNode / [HDFS](#)

✓ Metrics Monitor / [Ambari Metrics](#)

✓ NodeManager / [YARN](#)

Clients / HDFS Client, Hive Client,
Kerberos Client,
MapReduce2 Client, Oozie
Client, Pig Client

✓ impetus-i0163.impetus.co.in

← Back

Summary

Configs

Alerts 0

Versions

Components

✓ Metrics Collector / [Ambari Metrics](#)

✓ Grafana / [Ambari Metrics](#)

✓ ZooKeeper Server / [ZooKeeper](#)

✗ DataNode / [HDFS](#)

✓ SmartSense HST Agent / [SmartSense](#)

✓ JournalNode / [HDFS](#)

✓ Metrics Monitor / [Ambari Metrics](#)

✓ NodeManager / [YARN](#)

Clients / HDFS Client, Hive Client,
Kerberos Client,
MapReduce2 Client, Oozie
Client, Pig Client

During the decommission process the data present in *node* is dispersed to other *nodes*. Hadoop takes care of this procedure. Once the decommission process is completed we can safely remove the decommissioned node from the cluster.

Steps to commission a node:-

1. Add the network addresses of the new nodes to the include file in hdfs-site.xml

```
<property>
<name>dfs.hosts</name>
<value>/<hadoop-home>/conf/includes</value>
<final>true</final>
</property>
```

mapred-site.xml

```
<property>
<name>mapred.hosts</name>
<value>/<hadoop-home>/conf/includes</value>
<final>true</final>
</property>
```

Datanodes that are permitted to connect to the namenode are specified in a file whose name is specified by the dfs.hosts property. Includes file resides on the namenodes local filesystem, and it contains a line for each datanode, specified by network address (as reported by the datanode; you can see what this is by looking at the namenodes web UI). If you need to specify multiple network addresses for a datanode, put them on one line, separated by whitespace.

eg :

slave01
slave02
slave03

..

Similarly, tasktrackers that may connect to the jobtracker are specified in a file whose name is specified by the `mapred.hosts` property. In most cases, there is one shared file, referred to as the include file, that both `dfs.hosts` and `mapred.hosts` refer to, since nodes in the cluster run both datanode and tasktracker daemons.

2. Update the namenode with the new set of permitted datanodes using this command:

% `hadoop dfsadmin refreshNodes`

3. Update the job tracker with the new set of permitted task trackers using this command:

% `hadoop mradmin refreshNodes`

4. Update the slaves file with the new nodes, so that they are included in future operations performed by the Hadoop control scripts.

5. Start the new data nodes and task trackers.

6. Check that the new data nodes and task trackers appear in the web UI.