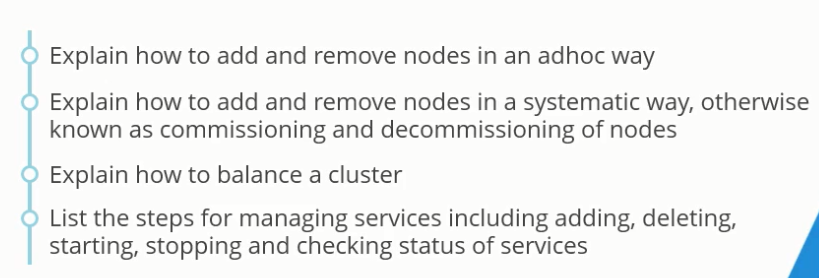
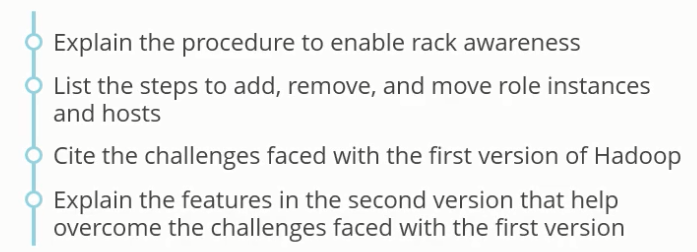
Cluster maintenance and administration

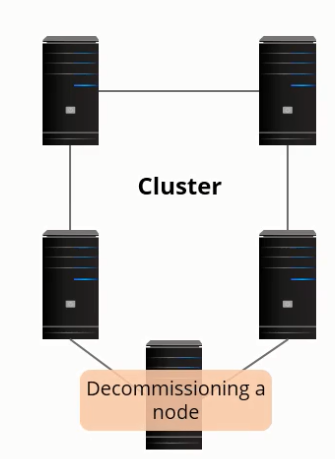
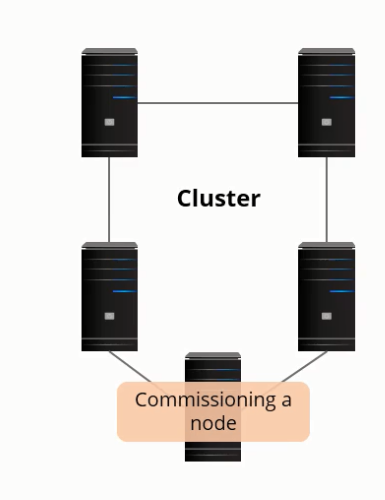


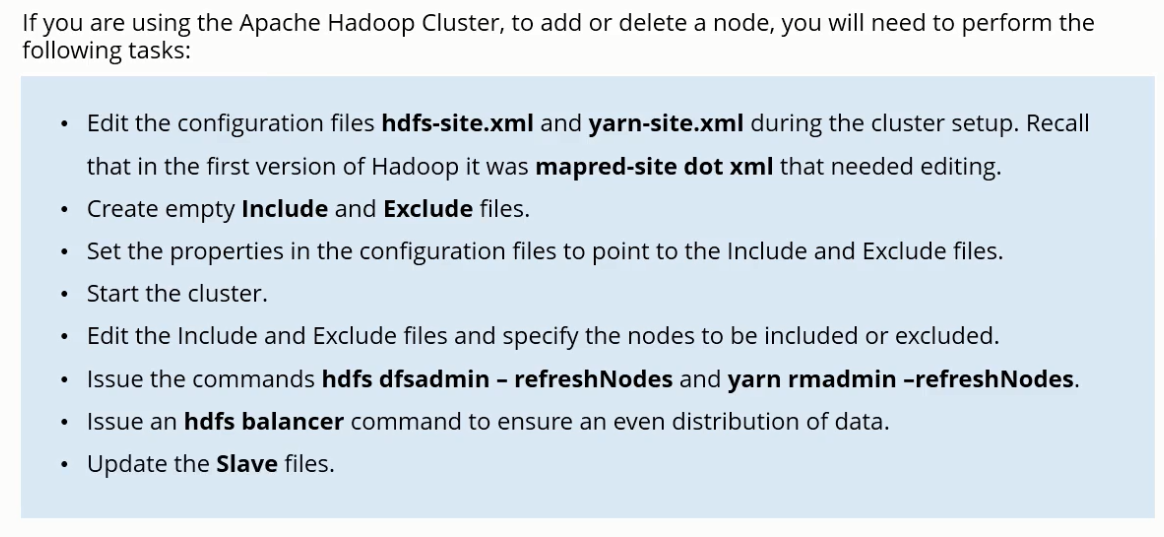


Add or remove nodes from a cluster

**Ad hoc method**: Allow the node to communicate with name node and simply remove the plug. This incurs a lot of overhead on the name node

**Systematic method**: While commissioning a node, inform the master node and YARN about it

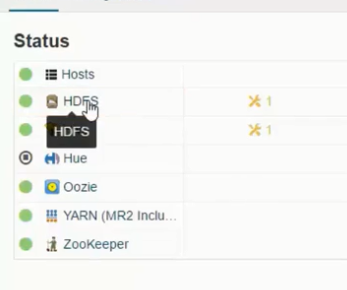




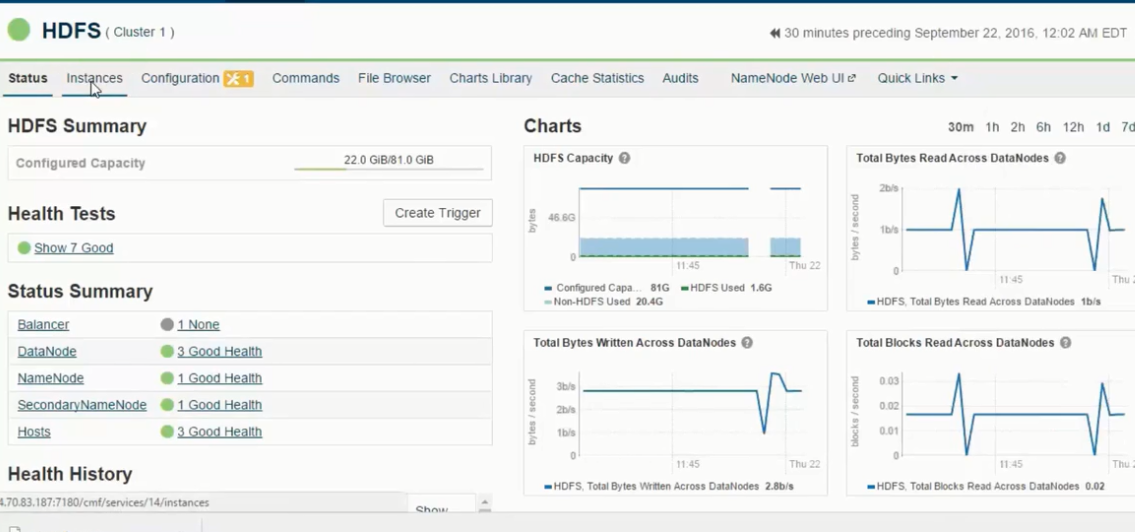
In addition, remove rack awareness if that has been implemented

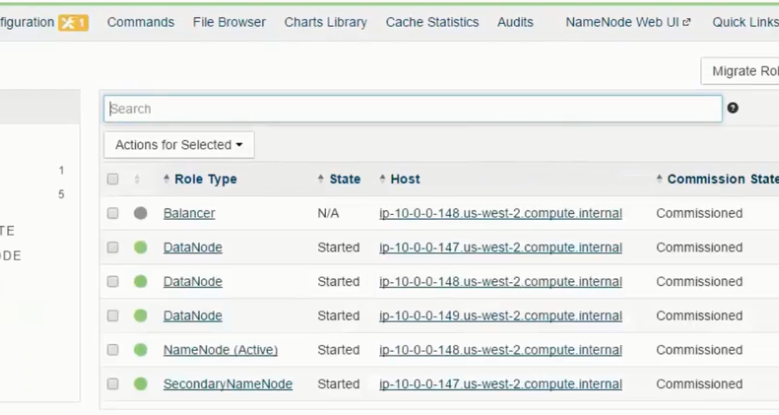
**Commissioning (adding a host or a machine to a cluster) and decommissioning a data node in Cloudera cluster**

Click on HDFS- shows 3 hosts running in the system



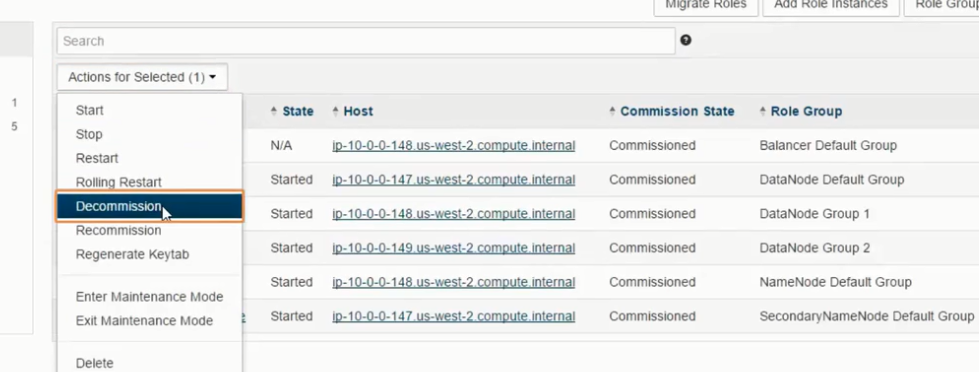
Click on instances





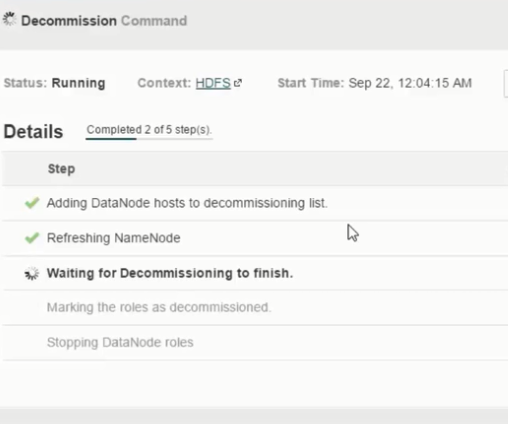
Decommission

Select the node to decommission (select the check box for the node), and choose decommission on the drop-down menu option labeled- Actions for Selected



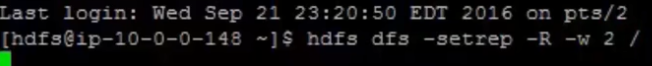
If the replication factor is 3, and there are 3 data nodes, then the process will not complete. In order to complete the process, alter the replication factor before proceeding to decommission the node.

Once the node is decommissioned, the cluster will not use the machine to store and process the data. If there are only 3 nodes with a replication factor of 3, then the decommissioning process will commence however for the process to complete, change the replication factor to 2 while the process is in progress- this process will take time due to under replicated nodes but the process will be successful once the replication factor has been modified.

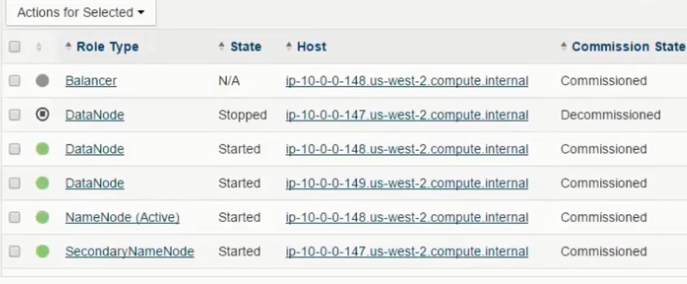


The data node is taking time as shown in the figure, as it is searching for one more node to replicate the blocks- replication factor is set to 3.

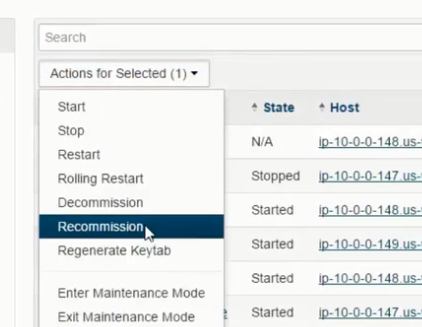
Changing the replication factor to 2 will complete the process:



The name node will delete one set of block- now the process of decommission is complete.



Recommission the machine – reverse the process- this adds the data nodes back to the cluster

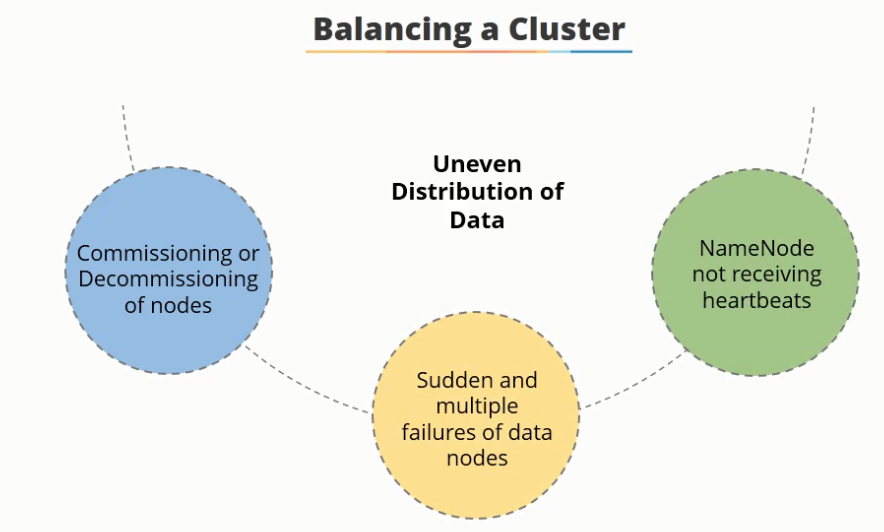




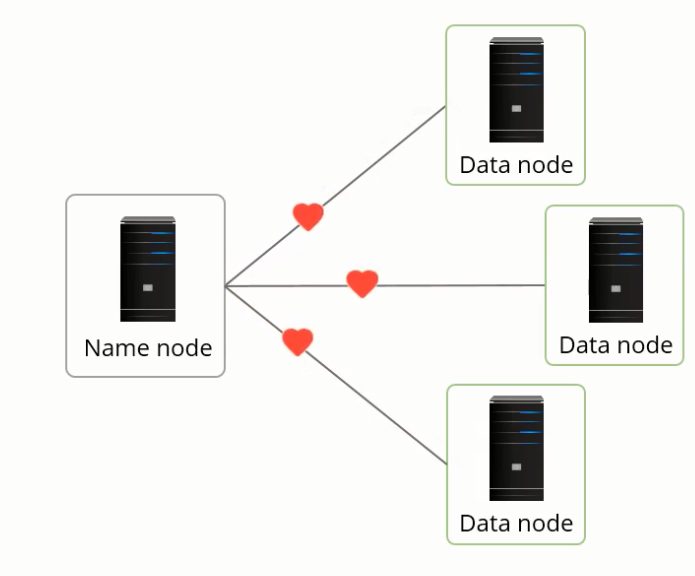
Start the recommissioned machine once the process is complete as the state is stopped

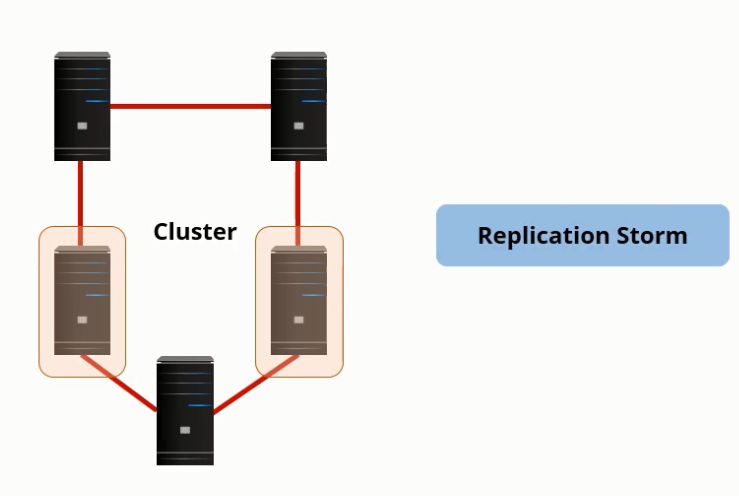


**Balancing a cluster**

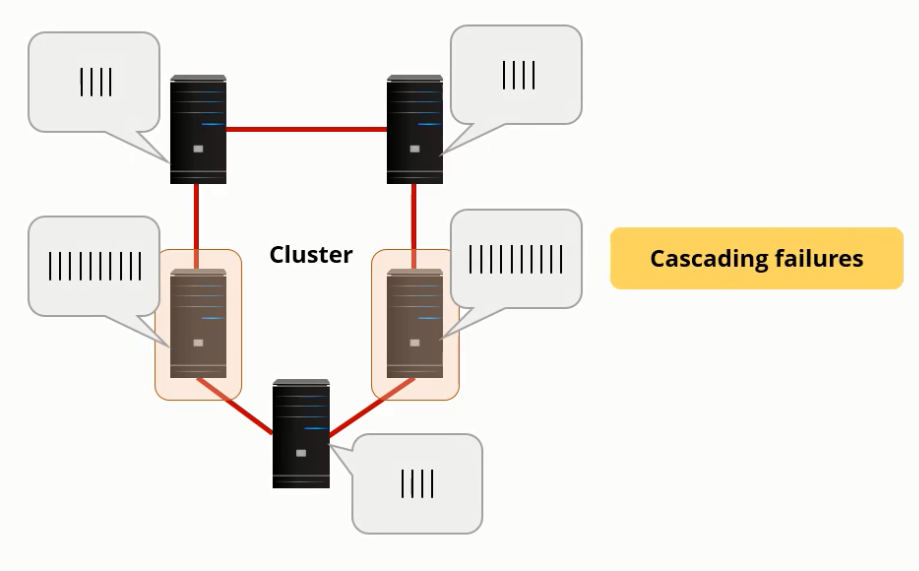


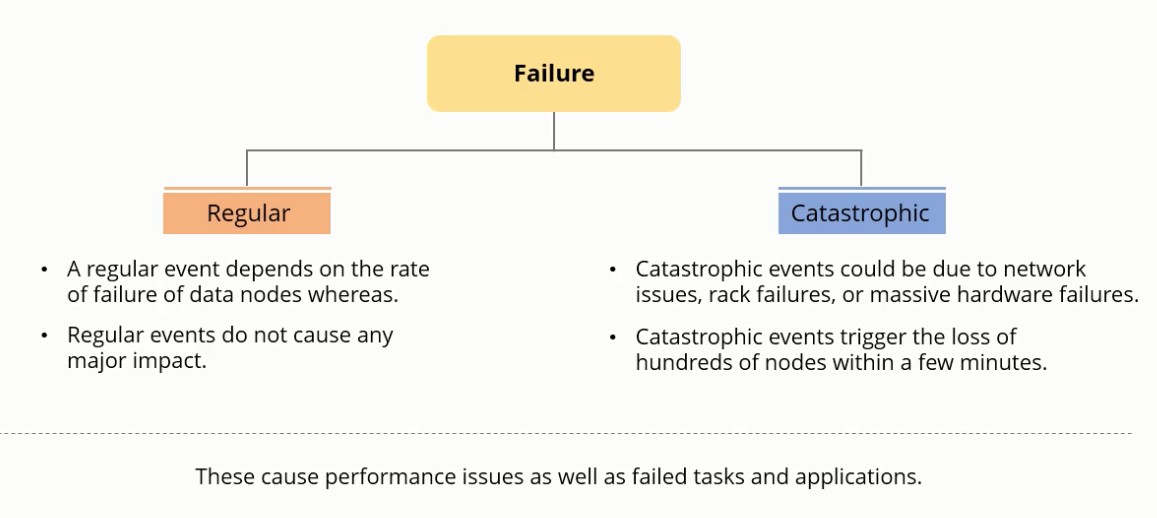
Due to a sudden failure of a data node or if the name node does not receive heartbeats from the data node, the, name node starts the process of data replication resulting in a replication storm





Moreover, if the failed data nodes contain more data than the other data nodes, it may result to cascading failures





HDFS Balancer helps to distribute data and avoid these problems. It balances by moving blocks from over utilized nodes to underutilized nodes. Useful when data nodes are commissioned or decommissioned, or when the data nodes fail suddenly, or even the name node stops receiving heartbeats from the data nodes.

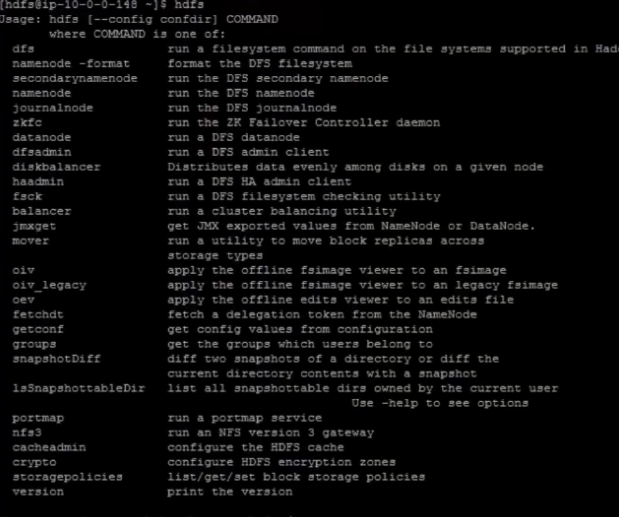
**Balance and re-balance data to ensure even distribution**

It is important to balance the data node after commissioning or decommissioning a node, or while adding or removing a node.

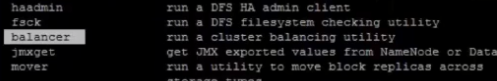
* Add a new cluster
* Deploy new client configurations
* Add role instances
* Commission and or add more data nodes

Balancing command for any running machine:

All of the hdfs commands are as followers (there is an option balancer)

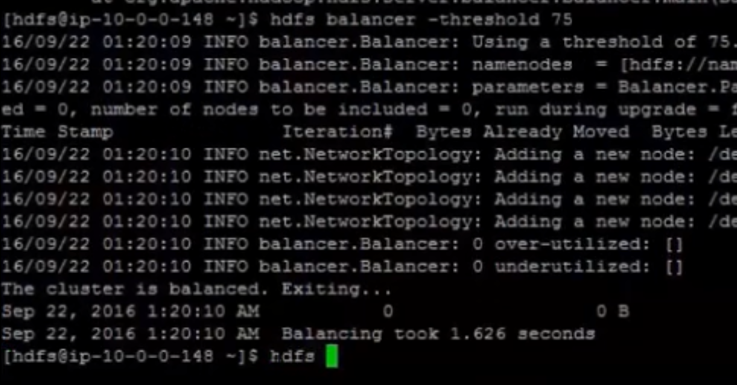


Balancer moves blocks from over utilized machines to underutilized machines





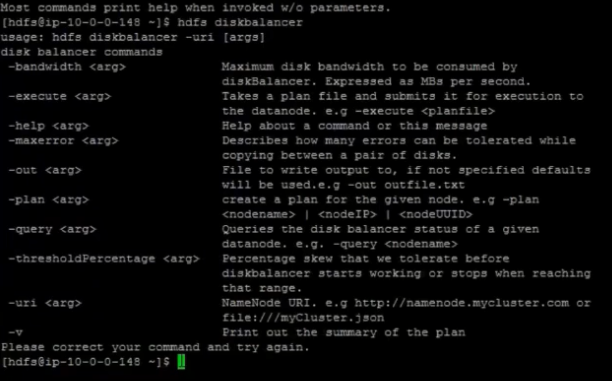
Threshold can also be specified with balancer- 75% is a good option. Load balancing is performed with the disk reaches 75%



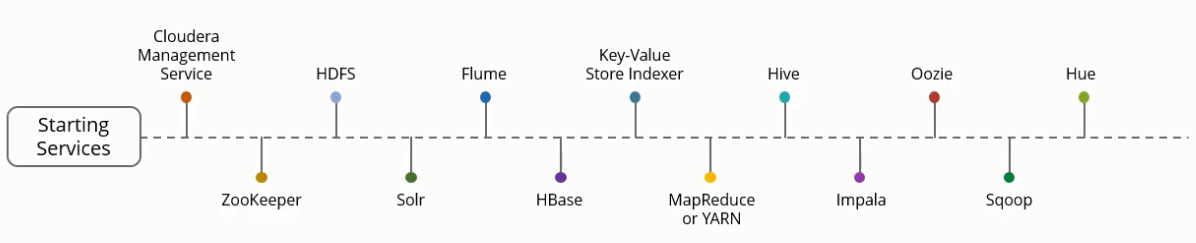
To see the number of data nodes working

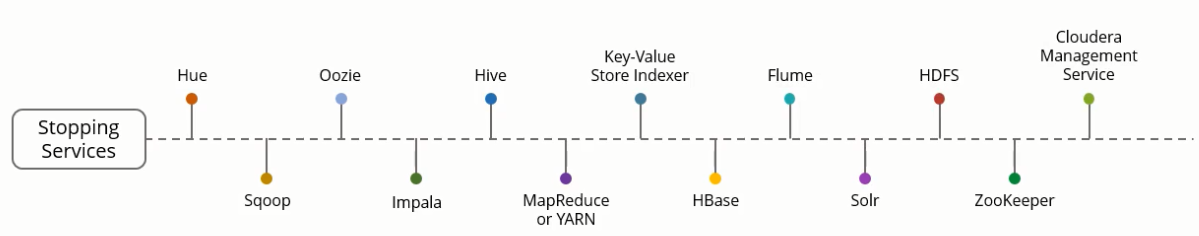


With multiple disks hdfs diskbalancer

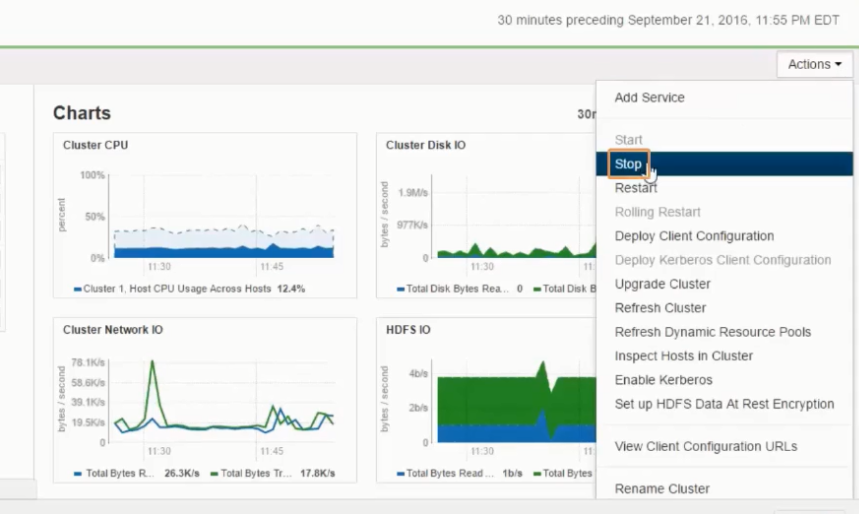


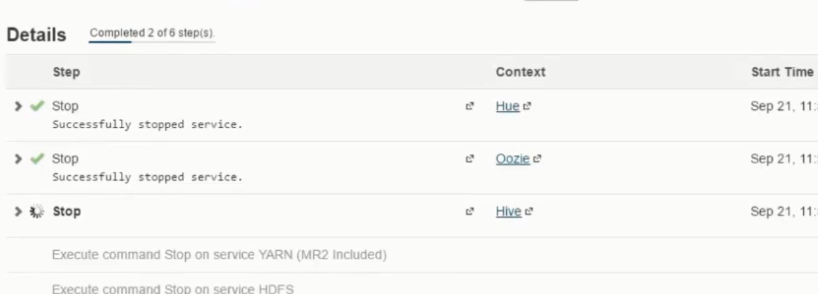
**Steps to start or start services within the Cloudera admin**



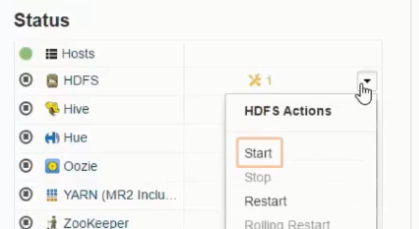


To stop all services

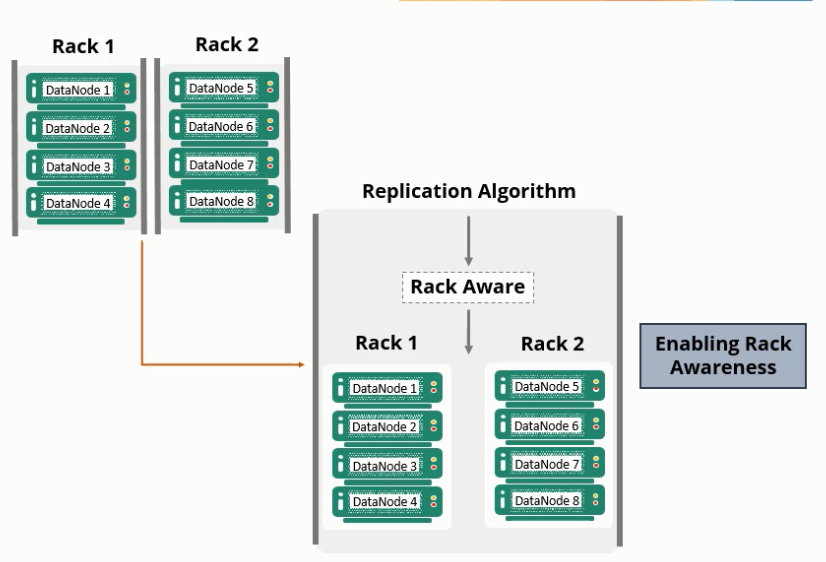




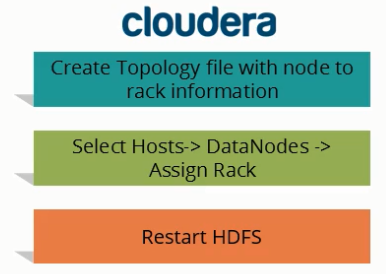
Starting individual services

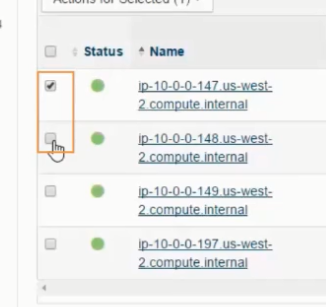
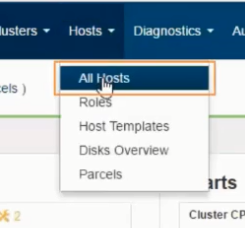


**Enabling Rack Awareness**

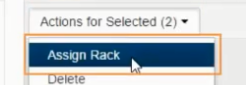


In Cloudera Manager and Hadoop



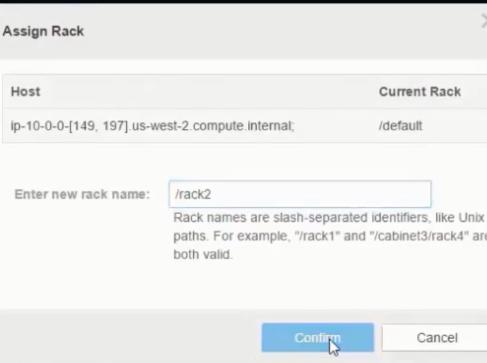
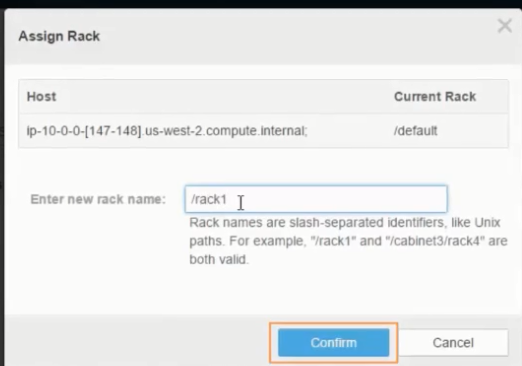


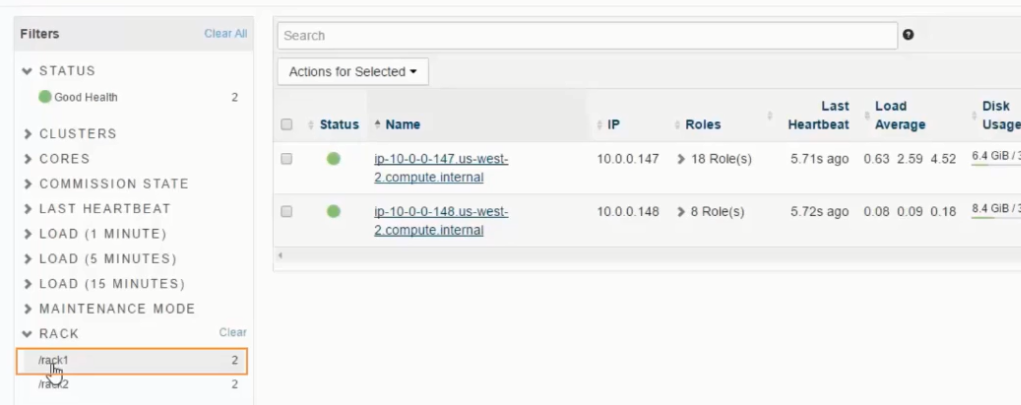
Select the hosts, and under action for selected click on assign rack



**Creating 2 racks**:

The first two nodes are assigned to rack 1 and the latter two are assigned to rack 2

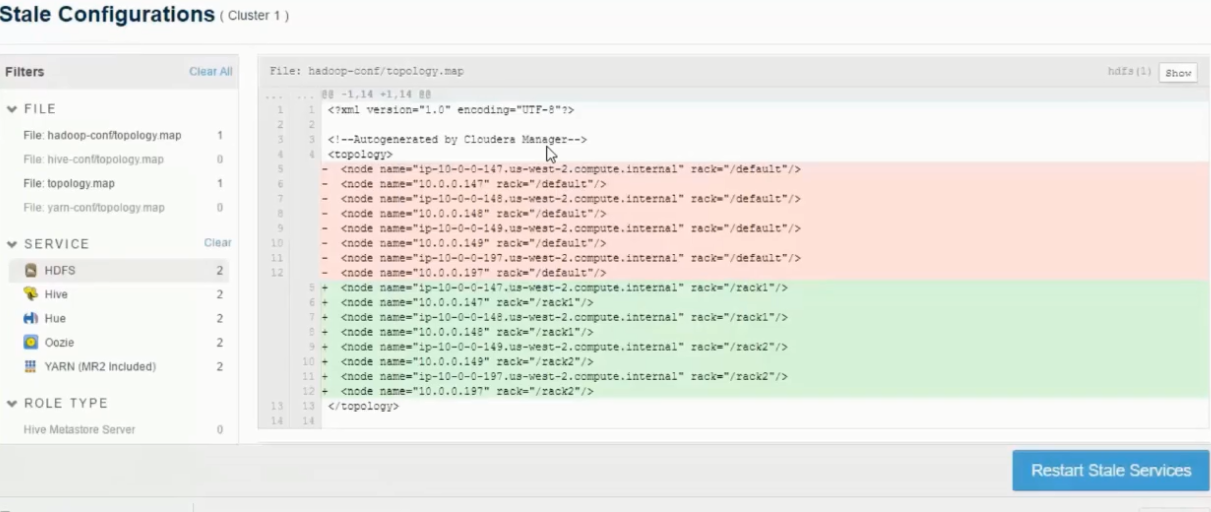




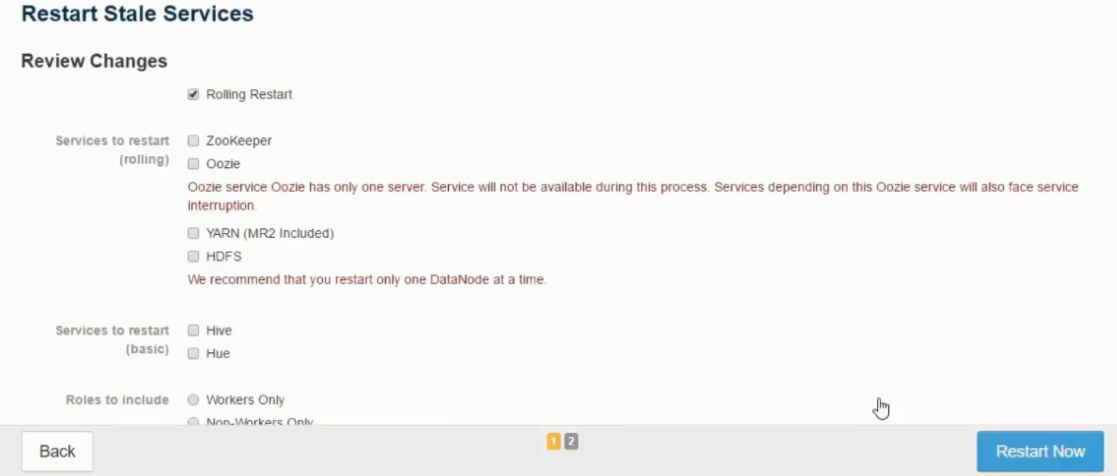
The changes need to be refreshed



Then restart stale services



And restart the service



Through command line



**Why do we need decommissioning and commissioning?**

You cannot directly remove any data node 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 data node/slave-nodes. Similarly, if you want to scale your cluster or add new data nodes without shutting down the cluster, you need commissioning.

**Factors affecting Commissioning and Decommissioning process:**

The first step is to contact the YARN manager. This is because it contains the records of all the running process. So, the first step is let YARN know that you are going to remove a data node and then you need to tell your Name node that you are going to remove a particular node. Next, let’s add the Decommissioning and Commissioning property into the core-site.xml file of the Master node (Name node).

**Steps for Decommissioning:**

1) Before add any property, stop your cluster. Otherwise, it will affect your cluster. You can do this using the command stop-dfs.sh

stop-dfs.sh

Next, go to your Resource Manager node to edit yarn-site.xml

2) You need to add this property in your yarn-site.xml

<property>

<name>yarn.resourcemanager.node.exclude-path</name>

<value>/home/hadoop/excludes</value>

</property>

Note- In value section, mention the excludes file address.

Now, go to your master node (Name node) and edit the hdfs-site.xml file-

3) Add this property to hdfs-site.xml

<property>

<name>dfs.hosts.exclude</name>

<value>/home/hadoop/excludes</value>

</property>

Note–If the Resource Manager and the Name node (Master Node) are on the same machine, then simply edit the yarn-site.xml and hdfs-site.xml of Name node (Master node)

4) Next, start your cluster using the following commands:

start-dfs.sh (Run this command On Master node/Name node only)

start-yarn.sh (Run this command On Resource Manager)

Note- If the Resource Manager (Node manager) and Name node are running on the same machine, then run the above commands on Name node (Master Node) only.

5) We need to update exclude file on both machine Resource manager and Name node (Master Node), if it’s not there then we can create an exclude file on both the machines

vi excludes

Add the Data node/Slave-node address, for decommissioning-

192.168.10.103

includes

6) Run the following command in the Resource Manager:

yarn rmadmin -refreshNodes (on Resource Manager)

This command will basically check the yarn-site.xml and process that property. And decommission the mentioned node from yarn. It means now yarn manager will not give any job to this node.

yarn rmadmin -refreshNodes

7) Run the following command on the Name node to check hdfs-site.xml and process the property and decommissioned the specified node/data node.

hdfs dfsadmin -refreshNodes

hdfs dfsadmin -refreshNodes (on Name node)

This command will basically check the yarn-site.xml and process that property, and Decommission the mentioned node from YARN. Meaning, the YARN Manager will not give any job to this node.

hadoop dfsadmin -refreshNodes

8) Run the command hadoop dfsadmin –report

hadoop dfsadmin -report

hadoop dfsadmin -report

hadoop dfsadmin -report

**Commissioning of Data nodes**:

Commissioning process is just the opposite of decommissioning, but the configuration part is almost same for both.

Follow the steps for commissioning configuration –

Before starting commissioning steps, simply remove the exclude file on both machine or delete all the entries of exclude file (make it blank)

Stop all daemons before adding any property into Hadoop cluster.

Open Resource manager machine to edit yarn-site.xml

1) Next, go to yarn manager, and add this property into yarn-site.xml.

vi yarn-site.xml

<property>

<name>yarn.resourcemanager.nodes.include-path</name>

<value>/home/hadoop/includes</value>

</property>

Next, Go to your Namenode (Master Node).

2) Add this property to hdfs-site.xml:

vi hdfs-site.xml (on Name node)

<property>

<name>dfs.hosts</name>

<value>/home/hadoop/includes</value>

</property>

3) Now, start your cluster using the following commands:

start-dfs.sh (Run this command On Name node only)

start-yarn.sh (Run this command On Resource Manager)

Note- If the Resource Manager (Node manager) and the Name node are running on same machine, then run these commands on Name node (Master Node) only.

4) We need to update the include file on both the Resource Manager and the Name node (Master Node). If it’s not present, then create an include file on both the Nodes.

vi includes

Add the Data node’s/Slave nodes IP address or hostname

192.168.10.101

192.168.10.102

192.168.10.103

exclude

Note- If you are going to add a new data node or if you are scaling up your cluster by adding new node, you need to add the IP address and hostname to /etc/hosts file of all nodes (Name node, Data node, Resource Manager).

Whenever you are going to do Commissioning, please mention all data node address in the Include file.

5) Run the following command on the Resource Manager

yarn rmadmin -refreshNodes (on Resource Manager)

yarn rmadmin -refreshNodes

6) Next, go to the Master Node (Namenode) and run the following command to refresh all nodes:

Run this command to refresh all nodes-

hdfs dfsadmin -refreshNodes

hadoop dfsadmin -refreshNodes

7) Check Hadoop admin report using the command hadoop dfsadmin –report.

hadoop dfsadmin -report

hadoop dfsadmin -report

Here, you can see that dn3.mycluster.com (192.168.10.103) data node, which was on decommissioned state, is now on the Normal state (Commissioned).

Note: The most important thing when you do commissioning is to make sure that the data node which you are going to add has everything (Should be configured for Hadoop data node).

And second thing which you need to keep in your mind is that, you should have to mention all necessary data nodes address in the include files.

Run cluster Balancer, as Balancer attempts to provide a balance to a certain threshold among data nodes by copying block data from older nodes to newly commissioned nodes.

hadoop balancer

Hadoop Balancer is a built-in property which makes sure that no data node will be over utilized. When you run the balancer utility, it checks whether some of the data nodes are under-utilized or over-utilized and will balance the replication factor. But make sure the Balancer should run in only off-peak hours in a real cluster, because if you run this during peak hours, it will cause a heavy load to networking, as it will transfer large amount of data.