Here is the steps of High Availability steps for 4 nodes cluster.

i am sure you won't find it anywhere in google search.

This blog is set up using the Quorum Journal Manager (QJM) to share and edit logs between the Active and Standby NameNodes.

In a typical HA cluster, two separate machines are configured as NameNodes. At any point of time, exactly one of the NameNodes is in an Active state, and the other is in a Standby state. The Active NameNode is responsible for all client operations in the cluster, while the Standby is simply acting as a slave, maintaining enough state to provide a fast failover if necessary.

To set up High Availability in Hadoop cluster you have to use Zookeeper in all the nodes.

The daemons in Active NameNode is:

Zookeeper

Zookeeper Fail Over controller

Journal Node

NameNode

The daemons in Standby Name node is:

Zookeeper

Zookeeper Fail Over controller

Journal Node

Name Node

The daemons in data node is:

Zookeeper

Journal Node

Data node

Zookeeper

Journal Node

Data node

You have to first set up the Java and host names of each node.

Virtual machines  IP address Host name

Active name node 192.168.0.32 namenode1

Standby namenode 192.168.0.33 namenode2

Data node        192.168.0.34 Datanode1

data node                192.168.0.35 Datanode2

vi /etc/hosts

make an entry in hosts file

[hduser@namenode1 opt]$ cat /etc/hosts

127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4

::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.0.32 namenode1

192.168.0.33 namenode2

192.168.0.34 datanode1

192.168.0.35 datanode2

yum install ntp ntpdate ntp-doc  ---(Install network time protocol all the nodes)

vi /etc/sysconfig/selinix       ----(Disabled on all nodes)

disabled

Run as Root User on all the nodes

sudo chkconfig ntpd on

sudo chkconfig iptables off

sudo /etc/init.d/iptables stop

sudo service ntpd start

sudo chkconfig --list ntpd

create user for hadoop

groupadd hadoop

useradd -g hadoop hduser

passwd hduser

ssh configuration on each node   ======

su - hduser

ssh-keygen -t rsa -P ""

cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

ssh-copy-id -i $HOME/.ssh/id\_rsa.pub hduser@namenode1

ssh-copy-id -i $HOME/.ssh/id\_rsa.pub hduser@namenode2

ssh-copy-id -i $HOME/.ssh/id\_rsa.pub hduser@datanode1

ssh-copy-id -i $HOME/.ssh/id\_rsa.pub hduser@datanode2

VISUDO (ROOT USER)     ====(On all nodes)

hduser  ALL=(ALL)       ALL

cd /opt

Download the Tarbal for Hadoop 2.6 and zookeeper 3.4 in /OPT or any other location and untar it.

wget <https://www.apache.org/dist/zookeeper/zookeeper-3.4.5/zookeeper-3.4.5.tar.gz>

wget <http://mirrors.advancedhosters.com/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz>

zookeeper-3.4.5.tar.gz

hadoop-2.6.0.tar.gz

make an Symbolic link

ln -s zookeeper-3.4.5.tar.gz  zookeeper

ln -s hadoop-2.6.0.tar.gz hadoop

Change the owner

Chown -R hduser:hadoop /opt/\*

Open the .bashrc file or .bash\_profile file   (on All the nodes)

Add the below paths:

export JAVA\_HOME=/usr/java/jdk1.8.0\_45

export PATH=$PATH:$JAVA\_HOME/bin

export HADOOP\_INSTALL=/opt/hadoop/

export PATH=$PATH:$HADOOP\_INSTALL/bin

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_HOME=$HADOOP\_INSTALL

export HADOOP\_CONF\_DIR=${HADOOP\_HOME}"/etc/hadoop"

Go to Zookeeper related configuration files are located

cd /opt/zookeeper/conf

make a copy of zoo\_sample.cfg to zoo.cfg and add below entries

cp zoo\_sample.cfg to zoo.cfg

vi zoo.cfg

tickTime=2000

clientPort=2181

initLimit=5

syncLimit=2

dataDir=/ha/zookeeper/data/

dataLogDir=/ha/zookeeper/logs/

server.1=namenode1:2888:3888

server.2=namenode2:2888:3888

server.3=datanode1:2888:3888

server.4=datanode2:2888:3888

save and exit

Create the folder structure for Zookeeper data and logs as defined in zoo.cfg in all the nodes

mkdir –p /ha/zookeeper/data/

mkdir –p /ha/zookeeper/logs/

chown -R hduser:hadoop /ha/zookeeper/\*

 Create the myid file in /ha/zookeeper/data/  in all nodes and assign the value of each of the nodes in cluster like namenode1 is the myid file is 1.

example below :

 vi myid

 1

 save and exit  (repeat steps in all other nodes with incremented number)

[hduser@namenode1:~]$vi /ha/zookeeper/data/myid

1

Save and Exit!

[hduser@namenode2~]$ vi /ha/zookeeper/data/myid

2

Save & Exit!

Hadoop configuration and high availability settings

cd /opt/hadoop/etc/hadoop

vi core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://ha-mycluster</value>

</property>

<property>

<name>dfs.journalnode.edits.dir</name>     ==== note:(Create the directory structure)

<value>/ha/journal/data</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/ha/journal/tmp</value> =====note:(Create the directory structure)

</property>

</configuration>

vi hdfs-site.xml

configuration>

<property>

 <name>dfs.namenode.name.dir</name>

 <value>/ha/data/namenode</value> ====note:(Create the directory structure)

 </property>

<property>

 <name>dfs.datanode.name.dir</name>

 <value>/ha/data/datanode</value>

 </property>

 <property>

 <name>dfs.replication</name>

 <value>2</value>

 </property>

 <property>

 <name>dfs.permissions</name>

 <value>false</value>

 </property>

 <property>

 <name>dfs.nameservices</name>

 <value>ha-mycluster</value>

 </property>

 <property>

 <name>dfs.ha.namenodes.ha-mycluster</name>

 <value>namenode1,namenode2</value>

 </property>

 <property>

 <name>dfs.namenode.rpc-address.ha-mycluster.namenode1</name>

 <value>namenode1:9000</value>

 </property>

 <property>

 <name>dfs.namenode.rpc-address.ha-mycluster.namenode2</name>

 <value>namenode2:9000</value>

 </property>

 <property>

 <name>dfs.namenode.http-address.ha-mycluster.namenode1</name>

 <value>namenode1:50070</value>

 </property>

 <property>

 <name>dfs.namenode.http-address.ha-mycluster.namenode2</name>

 <value>namenode2:50070</value>

 </property>

 <property>

 <name>dfs.namenode.shared.edits.dir</name>

 <value>qjournal://namenode1:8485;namenode2:8485;datanode1:8485;datanode2:8485/ha-mycluster</value>

 </property>

 <property>

 <name>dfs.client.failover.proxy.provider.ha-mycluster</name>

 <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

 </property>

 <property>

 <name>dfs.ha.automatic-failover.enabled</name>

 <value>true</value>

 </property>

 <property>

 <name>ha.zookeeper.quorum</name>

 <value>namenode1:2181,namenode2:2181,datanode1:2181,datanode2:2181</value>

 </property>

 <property>

 <name>dfs.ha.fencing.methods</name>

 <value>sshfence</value>

 </property>

 <property>

 <name>dfs.ha.fencing.ssh.private-key-files</name>

 <value>/home/hduser/.ssh/id\_rsa</value>

 </property>

</configuration>

vi yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

vi mapred-site.xml

 <configuration>

<property>

<name>[mapreduce.framework.name](http://mapreduce.framework.name/)</name>

<value>yarn</value>

</property>

</configuration>

vi hadoop-env.sh

export HADOOP\_HEAPSIZE=512

export JAVA\_HOME=/usr/java/jdk1.8.0\_45

vi yarn-env.sh

export JAVA\_HOME=/usr/java/jdk1.8.0\_45

vi slave    --(Put an entry on both the namenodes)

datanode1

datanode2

Start zookeeper service, once in all the nodes in cluster used for zookeeper, repeat below step in all the cluster nodes running zookeeper (

Go to zookepeer-3.4.5 Binary path i.e /opt/zookeeper/bin, then execute below commands

zkServer.sh start

You can check the status of zookeeper in all the nodes run in the command below. It will give the follower or Leader message.

[hduser@namenode1 hadoop]$ zkServer.sh status

JMX enabled by default

Using config: /opt/zookeeper/bin/../conf/zoo.cfg

**Mode: follower**

Start the Journalnode in all the 4 nodes.

Command: hadoop-daemon.sh start journalnode

check with JPS

Go to hadoop home path i.e /opt/hadoop/etc/hadoop, then execute below command

hdfs zkfc –formatZK

Format namenode

Go to hadoop home path i.e /opt/hadoop/etc/hadoop then execute below command

$bin/hdfs namenode –format

Copy Meta data information to slave name node in our guide (namenode2), run below command in

namenode2 (slave).

Make sure that namenode service is running in master node(n1)….

Go to hadoop home path i.e /opt/hadoop/etc/hadoop then execute below command

[sbin/hadoop-daemon.sh start namenode

Then in n

Go to hadoop home path i.e /home/cluster/hadoop-2.2.0-cdh5.0.0-beta-2, then execute below command

[cluster@n4~]$bin/hdfs namenode -bootstrapStandby

Finally check the active and standby status of the namenode1 and namenode2

[hduser@namenode1 hadoop]$ hdfs haadmin -getServiceState namenode1

16/03/21 04:36:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

**active**

[hduser@namenode1 hadoop]$ hdfs haadmin -getServiceState namenode2

16/03/21 04:37:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

**standby**