Hadoop full搭建

1. 基础设施：
2. Hadoop配置：
3. 规划角色：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | NN | JN | DN | ZKFC | ZK |
| Node01 | \* | \* |  | \* |  |
| Node02 | \* | \* | \* | \* | \* |
| Node03 |  | \* | \* | \* | \* |
| Node04 |  |  | \* |  | \* |

1. 规划路径：

mkdir /opt/bigdata

tar xzvf /root/soft/hadoop-2.6.5.tar.gz

mv hadoop-2.6.5 /opt/bigdata/

修改hadoop环境变量

vi /etc/profile

export HADOOP\_HOME=/opt/bigdata/hadoop-2.6.5

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

source /etc/profile

1. 配置hadoop的角色：

cd $HADOOP\_HOME/etc/hadoop

必须给hadoop配置javahome要不ssh过去找不到

vi hadoop-env.sh

export JAVA\_HOME=/usr/java/default

给出NN角色在哪里启动

vi core-site.xml

<property>

<name>fs.defaultFS</name>

<value>hdfs://node01:9000</value>

</property>

<property>

<name>ha.zookeeper.quorum</name>

<value>node02:2181,node03:2181,node04:2181</value>

</property>

vi hdfs-site.xml

配置hdfs 副本数为2.

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

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

<value>/var/bigdata/hadoop/ha/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/var/bigdata/hadoop/ha/dfs/data</value>

</property>

#一对多，逻辑到物理节点的映射

<property>

<name>dfs.nameservices</name>

<value>mycluster</value>

</property>

<property>

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

<value>nn1,nn2</value>

</property>

<property>

<name>dfs.namenode.rpc-address.mycluster.nn1</name>

<value>node01:8020</value>

</property>

<property>

<name>dfs.namenode.rpc-address.mycluster.nn2</name>

<value>node02:8020</value>

</property>

<property>

<name>dfs.namenode.http-address.mycluster.nn1</name>

<value>node01:50070</value>

</property>

<property>

<name>dfs.namenode.http-address.mycluster.nn2</name>

<value>node02:50070</value>

</property>

#JN服务配置，数据存储配置

<property>

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

<value>qjournal://node01:8485;node02:8485;node03:8485/mycluster</value>

</property>

<property>

<name>dfs.journalnode.edits.dir</name>

<value>/var/bigdata/hadoop/ha/dfs/jn</value>

</property>

#HA角色切换的代理类和实现方法，我们用的ssh免密

<property>

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

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

</property>

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

<property>

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

<value>/root/.ssh/id\_dsa</value>

</property>

#开启自动化：启动zkfc

<property>

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

<value>true</value>

</property>

配置DN这个角色再那里启动

vi slaves

node02

node03

node04

分发配置

scp hdfs-site.xml core-stie.xml node02:`pwd`

scp hdfs-site.xml core-stie.xml node03:`pwd`

scp hdfs-site.xml core-stie.xml node04:`pwd`

1. Zookeeper配置：

node02:

tar xf zook....tar.gz

mv zoo... /opt/bigdata

cd /opt/bigdata/zoo....

cd conf

cp zoo\_sample.cfg zoo.cfg

vi zoo.cfg

datadir=/var/bigdata/hadoop/zk

server.1=node02:2888:3888

server.2=node03:2888:3888

server.3=node04:2888:3888

mkdir /var/bigdata/hadoop/zk

echo 1 > /var/bigdata/hadoop/zk/myid

vi /etc/profile

export ZOOKEEPER\_HOME=/opt/bigdata/zookeeper-3.4.6

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$ZOOKEEPER\_HOME/bin

. /etc/profile

scp /etc/profile node03:/etc/

scp /etc/profile node04:/etc/

cd /opt/bigdata

scp -r ./zookeeper-3.4.6 node03:`pwd`

scp -r ./zookeeper-3.4.6 node04:`pwd`

node03:

mkdir /var/bigdata/hadoop/zk

echo 2 > /var/bigdata/hadoop/zk/myid

. /etc/profile

node04:

mkdir /var/bigdata/hadoop/zk

echo 3 > /var/bigdata/hadoop/zk/myid

. /etc/profile

1. 初始化&启动：

启动zookeeper

Node02-node04:

zkServer.sh start

初始化启动hadoop

1. node01-node03启动JN

hadoop-daemon.sh start journalnode

1. node01格式化NN

hdfs namenode -format

1. 启动格式化的NN

hadoop-daemon.sh start namenode

1. node02

hdfs namenode -bootstrapStandby

1. 格式化zk：

hdfs zkfc -formatZK

1. 正常启动：

start-dfs.sh

1. 使用验证

1）去看jn的日志和目录变化：

2）node04

zkCli.sh

ls /

启动之后可以看到锁：

get /hadoop-ha/mycluster/ActiveStandbyElectorLock

3）杀死namenode 杀死zkfc

kill -9 xxx

a)杀死active NN

b)杀死active NN身边的zkfc

c)shutdown activeNN 主机的网卡 ： ifconfig eth0 down

2节点一直阻塞降级

如果恢复1上的网卡 ifconfig eth0 up

最终 2编程active