

保持自己的节奏前进就对了

HDFS-HA集群配置

(一)环境准备

- 修改IP
- 修改主机名及主机名和IP地址的映射
- 关闭防火墙
- ssh免密登录
- 安装JDK，配置环境变量等

(二)规划集群

看出我们将NameNode分布到两台机器上，保证集群的高可用性

hadoop102	hadoop103	hadoop104
NameNode	NameNode	
JournalNode	JournalNode	JournalNode
DataNode	DataNode	DataNode
ZK	ZK	ZK
	ResourceManager	
NodeManager	NodeManager	NodeManager

(三)配置Zookeeper集群

1.集群规划

在hadoop102、hadoop103和hadoop104三个节点上部署Zookeeper。

2.解压安装

(1) 解压Zookeeper安装包到/opt/module/目录下

```
tar -zxvf zookeeper-3.4.10.tar.gz -C /opt/module/
```

(2) 在/opt/module/zookeeper-3.4.10/这个目录下创建zkData

```
mkdir -p zkData
```

(3) 重命名/opt/module/zookeeper-3.4.10/conf这个目录下的zoo_sample.cfg为zoo.cfg

```
mv zoo_sample.cfg zoo.cfg
```

3.配置zoo.cfg文件

(1) 具体配置

```
dataDir=/opt/module/zookeeper-3.4.10/zkData
```

增加如下配置

```
\#####cluster#####
```

```
server.2=hadoop102:2888:3888
```

```
server.3=hadoop103:2888:3888
```

```
server.4=hadoop104:2888:3888
```

(2) 配置参数解读

Server.A=B:C:D。

A是一个数字，表示这个是第几号服务器；

B是这个服务器的**IP**地址；

C是这个服务器与集群中的**Leader**服务器交换信息的端口；

D是万一集群中的**Leader**服务器挂了，需要一个端口来重新进行选举，选出一个新的**Leader**，而这个端口就是用来执行选举时服务器相互通信的端口。

集群模式下配置一个文件**myid**，这个文件在**dataDir**目录下，这个文件里面有一个数据就是**A**的值，**zookeeper**启动时读取此文件，拿到里面的数据与**zoo.cfg**里面的配置信息比较从而判断到底是哪个**server**。

4.集群操作

(1) 在/opt/module/zookeeper-3.4.10/zkData目录下创建一个myid的文件

```
touch myid
```

添加**myid**文件，注意一定要在**linux**里面创建，在**notepad++**里面很可能乱码

(2) 编辑myid文件

```
vi myid
```

在文件中添加与**server**对应的编号：如2

(3) 拷贝配置好的zookeeper到其他机器上

```
scp -r zookeeper-3.4.10/ [root@hadoop103.atguigu.com:/opt/app/]  
(mailto:root@hadoop103.atguigu.com:/opt/app/)
```

```
scp -r zookeeper-3.4.10/ [root@hadoop104.atguigu.com:/opt/app/]  
(mailto:root@hadoop104.atguigu.com:/opt/app/)
```

并分别修改**myid**文件中内容为3、4

(4) 分别启动zookeeper

```
[root@hadoop102 zookeeper-3.4.10]# bin/zkServer.sh start

[root@hadoop103 zookeeper-3.4.10]# bin/zkServer.sh start

[root@hadoop104 zookeeper-3.4.10]# bin/zkServer.sh start
```

(5) 查看状态

```
[root@hadoop102 zookeeper-3.4.10]# bin/zkServer.sh status
JMX enabled by default
Using config: /opt/module/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower

[root@hadoop103 zookeeper-3.4.10]# bin/zkServer.sh status
JMX enabled by default
Using config: /opt/module/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: leader

[root@hadoop104 zookeeper-3.4.5]# bin/zkServer.sh status
JMX enabled by default
Using config: /opt/module/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
```

(四)配置HDFS-HA集群

1.官方地址

```
http://hadoop.apache.org/
```

2.在opt目录下创建一个ha文件夹

```
mkdir ha
```

3.将/opt/app/下的 hadoop-2.7.2拷贝到/opt/ha目录下

```
cp -r hadoop-2.7.2/ /opt/ha/
```

4.配置hadoop-env.sh

```
export JAVA_HOME=/opt/module/jdk1.8.0_144
```

5.配置core-site.xml

```

<configuration>
<!-- 把两个NameNode的地址组装成一个集群mycluster -->
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://mycluster</value>
    </property>

    <!-- 指定hadoop运行时产生文件的存储目录 -->
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/opt/ha/hadoop-2.7.2/data/tmp</value>
    </property>
</configuration>

```

6.配置hdfs-site.xml

```

<configuration>
    <!-- 完全分布式集群名称 -->
    <property>
        <name>dfs.nameservices</name>
        <value>mycluster</value>
    </property>

    <!-- 集群中NameNode节点都有哪些 -->
    <property>
        <name>dfs.ha.namenodes.mycluster</name>
        <value>nn1,nn2</value>
    </property>

    <!-- nn1的RPC通信地址 -->
    <property>
        <name>dfs.namenode.rpc-address.mycluster.nn1</name>
        <value>hadoop102:9000</value>
    </property>

    <!-- nn2的RPC通信地址 -->
    <property>
        <name>dfs.namenode.rpc-address.mycluster.nn2</name>
        <value>hadoop103:9000</value>
    </property>

    <!-- nn1的http通信地址 -->
    <property>
        <name>dfs.namenode.http-address.mycluster.nn1</name>
        <value>hadoop102:50070</value>
    </property>

    <!-- nn2的http通信地址 -->
    <property>
        <name>dfs.namenode.http-address.mycluster.nn2</name>
        <value>hadoop103:50070</value>
    </property>

    <!-- 指定NameNode元数据在JournalNode上的存放位置 -->
    <property>
        <name>dfs.namenode.shared.edits.dir</name>

```

```

<value>qjournal://hadoop102:8485;hadoop103:8485;hadoop104:8485/mycluster</value>
</property>

<!-- 配置隔离机制，即同一时刻只能有一台服务器对外响应 -->
<property>
    <name>dfs.ha.fencing.methods</name>
    <value>sshfence</value>
</property>

<!-- 使用隔离机制时需要ssh无秘钥登录-->
<property>
    <name>dfs.ha.fencing.ssh.private-key-files</name>
    <value>/home/zhutiansama/.ssh/id_rsa</value>
</property>

<!-- 声明journalnode服务器存储目录-->
<property>
    <name>dfs.journalnode.edits.dir</name>
    <value>/opt/ha/hadoop-2.7.2/data/jn</value>
</property>

<!-- 关闭权限检查-->
<property>
    <name>dfs.permissions.enable</name>
    <value>false</value>
</property>

<!-- 访问代理类: client, mycluster, active配置失败自动切换实现方式-->
<property>
    <name>dfs.client.failover.proxy.provider.mycluster</name>

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

```

7.拷贝配置好的hadoop环境到其他节点

(五)启动HDFS-HA集群

1.在各个JournalNode节点上，输入以下命令启动journalnode服务

```
sbin/hadoop-daemon.sh start journalnode
```

2.在[nn1]上，对其进行格式化，并启动

```
bin/hdfs namenode -format
sbin/hadoop-daemon.sh start namenode
```

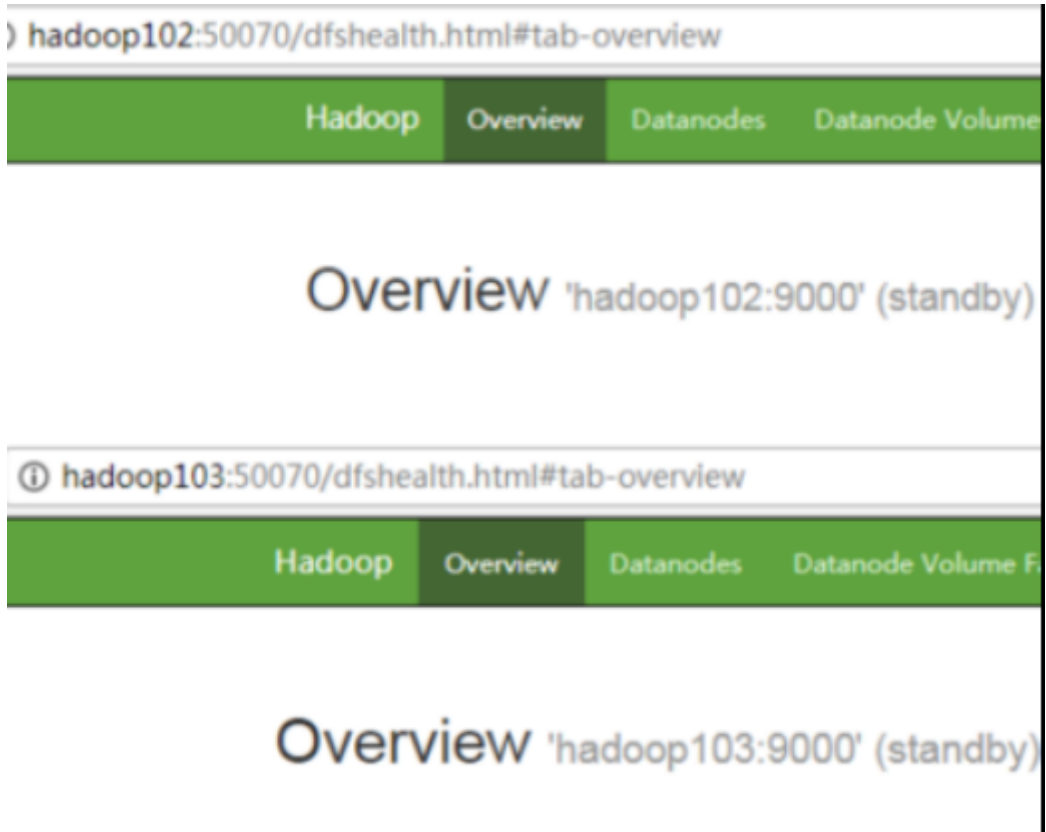
3.在[nn2]上，同步nn1的元数据信息

```
bin/hdfs namenode -bootstrapStandby
```

4.启动[nn2]

```
sbin/hadoop-daemon.sh start namenode
```

5.Web页面查看



6.在[nn1]上，启动所有datanode

```
sbin/hadoop-daemons.sh start datanode
```

7.将[nn1]切换为Active

```
bin/hdfs haadmin -transitionToActive nn1
```

8.查看是否Active

```
bin/hdfs haadmin -getServiceState nn1
```

(六)配置HDFS-HA自动故障转移

1.具体配置

(1) 在hdfs-site.xml中增加

```
<property>
  <name>dfs.ha.automatic-failover.enabled</name>
  <value>true</value>
</property>
```

(2) 在core-site.xml文件中增加

```
<property>
  <name>ha.zookeeper.quorum</name>
  <value>hadoop102:2181,hadoop103:2181,hadoop104:2181</value>
</property>
```

2.启动

(1) 关闭所有HDFS服务:

```
sbin/stop-dfs.sh
```

(2) 启动Zookeeper集群:

```
bin/zkServer.sh start
```

(3) 初始化HA在Zookeeper中状态:

```
bin/hdfs zkfc -formatzk
```

(4) 启动HDFS服务:

```
sbin/start-dfs.sh
```

(5) 在各个NameNode节点上启动DFSZK Failover Controller, 先在哪台机器启动, 哪个机器的NameNode就是Active NameNode

```
sbin/hadoop-daemon.sh start zkfc
```

3.验证

(1) 将Active NameNode进程kill

```
kill -9 namenode的进程id
```

(2) 将Active NameNode机器断开网络

```
service network stop
```