高可用hadoop集群搭建

3台服务架构

机器名	服务	作用
hd01	NameNode(主)	active状态名节点
	DFSZKFailoverController	故障自动转移
	JournalNode	Namenode数据同步
	DataNode	数据节点
	NodeManager	节点管理
	jobHistoryServer	历史服务
	chronyd服务	时间同步
	zookeeper(QuorumPeerMain)/JDK	ZOOKEEPER
hd02	NameNode(备)	备份名节点
	DFSZKFailoverController	故障自动转移
	JournalNode	Namenode数据同步
	DataNode	数据节点
	NodeManager	节点管理
	ResourceManager(备)	备份资源管理
	chronyd服务	时间同步
	zookeeper(QuorumPeerMain)/JDK	ZOOKEEPER
hd03	JournalNode	Namenode数据同步
	DataNode	数据节点
	NodeManager	节点管理
	ResourceManager(主)	主资源管理进程
	chronyd服务	时间同步
	zookeeper(QuorumPeerMain)/JDK	ZOOKEEPER

准备工作

1. 安装3台centos7 服务器
2. 配置名字hd01\hd02\hd03
3. 配置网络static
4. 关闭防火墙

5. hd01可以访问hd02、hd03（同理其他都可互相访问）

```
#hd01 做ssh 公私钥 无秘
ssh-keygen -t rsa -P ''
# copy 公钥到 hd02 hd03
ssh-copy-id 192.168.192.201#hd02
ssh-copy-id 192.168.192.202#hd03
```

6. 所有服务器时间同步

```
# 安装chrony
yum -y install chrony
#配置chrony
vi /etc/chrony.conf
server ntp1.aliyun.com
server ntp2.aliyun.com
server ntp3.aliyun.com
注释掉server 0.centos.pool.ntp.org iburst
#启动chrony
systemctl start chronyd
```

7. 安装wget

```
yum install -y wget
```

8. 安装psmisc(linux命令工具包 namenode主备切换时要用到 只需要安装在两个namenode节点上)

```
yum install -y psmisc
```

9. 修改yum源(阿里云)

```
# 进入阿里云镜像站点
https://developer.aliyun.com/mirror/ #这里有阿里云的指导
#备份原始源
mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-
Base.repo.backup
#下载源
wget -O /etc/yum.repos.d/CentOS-Base.repo
https://mirrors.aliyun.com/repo/Centos-7.repo
#清除缓存
yum clean all      # 清除系统所有的yum缓存
yum makecache      # 生成yum缓存
```

集群部署

1. 安装JDK
2. 安装zookeeper集群

```
# 配置zoo.cfg
dataDir=/opt/soft/zk345/data
server.1=hd01:2888:3888
server.2=hd01:2888:3888
server.3=hd01:2888:3888
#在每个dataDir路径下建立一个myid文件(不同主机的数字同上)
echo "1"> myid
#启动集群
zkServer.sh start
```

3. 安装hadoop集群

    1. 在单台上配置1个hadoop环境 创建文件夹

    ```
    # 解压
    tar -zxf hadoop-2.6.0-cdh5.14.2.tar.gz
    # 移动到自己的安装文件夹下
    mv hadoop-2.6.0-cdh5.14.2 soft/hadoop260
    # 添加对应各个文件夹
    mkdir -p /opt/soft/hadoop260/tmp
    mkdir -p /opt/soft/hadoop260/dfs/journalnode_data
    mkdir -p /opt/soft/hadoop260/dfs/edits
    mkdir -p /opt/soft/hadoop260/dfs/datanode_data
    mkdir -p /opt/soft/hadoop260/dfs/namenode_data
    ```

    2. 配置hadoop-env.sh

    ```
    export JAVA_HOME=/opt/soft/jdk180
    export HADOOP_CONF_DIR=/opt/soft/hadoop260/etc/hadoop
    ```

    3. 配置core-site.xml

    ```
    <configuration>
        <!--指定hadoop集群在zookeeper上注册的节点名-->
        <property>
            <name>fs.defaultFS</name>
            <value>hdfs://hacluster</value>
        </property>
        <!--指定hadoop运行时产生的临时文件-->
        <property>
            <name>hadoop.tmp.dir</name>
            <value>file:///opt/soft/hadoop260/tmp</value>
        </property>
        <!--设置缓存大小 默认4KB-->
        <property>
            <name>io.file.buffer.size</name>
            <value>4096</value>
        </property>
        <!--指定zookeeper的存放地址-->
        <property>
            <name>ha.zookeeper.quorum</name>
            <value>hd01:2181,hd02:2181,hd03:2181</value>
        </property>
        <!--配置允许root代理访问主机节点-->
        <property>
            <name>hadoop.proxyuser.root.hosts</name>
    ```

```
        <value>*</value>
    </property>
    <!--配置该节点允许root用户所属的组-->
    <property>
        <name>hadoop.proxyuser.root.groups</name>
        <value>*</value>
    </property>
</configuration>
```

4. 配置hdfs-site.xml

```
<configuration>
    <property>
        <!--数据块默认大小128M-->
        <name>dfs.block.size</name>
        <value>134217728</value>
    </property>
    <property>
        <!--副本数量  不配置默认为3-->
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <!--namenode节点数据(元数据)的存放位置-->
        <name>dfs.name.dir</name>
        <value>file:///opt/soft/hadoop260/dfs/namenode_data</value>
    </property>
    <property>
        <!--datanode节点数据(元数据)的存放位置-->
        <name>dfs.data.dir</name>
        <value>file:///opt/soft/hadoop260/dfs/datanode_data</value>
    </property>
    <property>
        <!--开启hdfs的webui界面-->
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
    </property>
    <property>
        <!--datanode上负责进行文件操作的线程数-->
        <name>dfs.datanode.max.transfer.threads</name>
        <value>4096</value>
    </property>
    <property>
        <!--指定hadoop集群在zookeeper上的注册名-->
        <name>dfs.nameservices</name>
        <value>hacluster</value>
    </property>
    <property>
        <!--hacluster集群下有两个namenode分别是nn1,nn2-->
        <name>dfs.ha.namenodes.hacluster</name>
        <value>nn1,nn2</value>
    </property>
    <!--nn1的rpc、servicepc和http通讯地址 -->
    <property>
        <name>dfs.namenode.rpc-address.hacluster.nn1</name>
        <value>hd01:9000</value>
    </property>
```

```
    <property>
        <name>dfs.namenode.servicepc-address.hacluster.nn1</name>
        <value>hd01:53310</value>
    </property>
    <property>
        <name>dfs.namenode.http-address.hacluster.nn1</name>
        <value>hd01:50070</value>
    </property>
    <!--nn2的rpc、servicepc和http通讯地址 -->
    <property>
        <name>dfs.namenode.rpc-address.hacluster.nn2</name>
        <value>hd02:9000</value>
    </property>
    <property>
        <name>dfs.namenode.servicepc-address.hacluster.nn2</name>
        <value>hd02:53310</value>
    </property>
    <property>
        <name>dfs.namenode.http-address.hacluster.nn2</name>
        <value>hd02:50070</value>
    </property>
    <property>
        <!--指定Namenode的元数据在JournalNode上存放的位置-->
        <name>dfs.namenode.shared.edits.dir</name>

 <value>qjournal://hd01:8485;hd02:8485;hd03:8485/hacluster</value>
    </property>
    <property>
        <!--指定JournalNode在本地磁盘的存储位置-->
        <name>dfs.journalnode.edits.dir</name>
        <value>/opt/soft/hadoop260/dfs/journalnode_data</value>
    </property>
    <property>
        <!--指定namenode操作日志存储位置-->
        <name>dfs.namenode.edits.dir</name>
        <value>/opt/soft/hadoop260/dfs/edits</value>
    </property>
    <property>
        <!--开启namenode故障转移自动切换-->
        <name>dfs.ha.automatic-failover.enabled</name>
        <value>true</value>
    </property>
    <property>
        <!--配置失败自动切换实现方式-->
        <name>dfs.client.failover.proxy.provider.hacluster</name>

 <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProx
yProvider</value>
    </property>
    <property>
        <!--配置隔离机制-->
        <name>dfs.ha.fencing.methods</name>
        <value>sshfence</value>
    </property>
    <property>
        <!--配置隔离机制需要SSH免密登录-->
        <name>dfs.ha.fencing.ssh.private-key-files</name>
        <value>/root/.ssh/id_rsa</value>
```

```
    </property>
    <property>
        <!--hdfs文件操作权限 false为不验证-->
        <name>dfs.premissions</name>
        <value>false</value>
    </property>
</configuration>
```

5. 配置mapper-site.xml

```
<configuration>
    <property>
        <!--指定mapreduce在yarn上运行-->
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <!--配置历史服务器地址-->
        <name>mapreduce.jobhistory.address</name>
        <value>hd01:10020</value>
    </property>
    <property>
        <!--配置历史服务器webUI地址-->
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>hd01:19888</value>
    </property>
    <property>
        <!--开启uber模式-->
        <name>mapreduce.job.ubertask.enable</name>
        <value>true</value>
    </property>
</configuration>
```

6. 配置yarn-site.xml

```
<configuration>
    <property>
        <!--开启yarn高可用-->
        <name>yarn.resourcemanager.ha.enabled</name>
        <value>true</value>
    </property>
    <property>
        <!-- 指定Yarn集群在zookeeper上注册的节点名-->
        <name>yarn.resourcemanager.cluster-id</name>
        <value>hayarn</value>
    </property>
    <property>
        <!--指定两个resourcemanager的名称-->
        <name>yarn.resourcemanager.ha.rm-ids</name>
        <value>rm1,rm2</value>
    </property>
    <property>
        <!--指定rm1的主机-->
        <name>yarn.resourcemanager.hostname.rm1</name>
        <value>hd02</value>
    </property>
```

```
    <property>
        <!--指定rm2的主机-->
        <name>yarn.resourcemanager.hostname.rm2</name>
        <value>hd03</value>
    </property>
    <property>
        <!--配置zookeeper的地址-->
        <name>yarn.resourcemanager.zk-address</name>
        <value>hd01:2181,hd02:2181,hd03:2181</value>
    </property>
    <property>
        <!--开启yarn恢复机制-->
        <name>yarn.resourcemanager.recovery.enabled</name>
        <value>true</value>
    </property>
    <property>
        <!--配置执行resourcemanager恢复机制实现类-->
        <name>yarn.resourcemanager.store.class</name>

 <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMState
Store</value>
    </property>
    <property>
        <!--指定主resourcemanager的地址-->
        <name>yarn.resourcemanager.hostname</name>
        <value>hd03</value>
    </property>
    <property>
        <!--nodemanager获取数据的方式-->
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <!--开启日志聚集功能-->
        <name>yarn.log-aggregation-enable</name>
        <value>true</value>
    </property>
    <property>
        <!--配置日志保留7天-->
        <name>yarn.log-aggregation.retain-seconds</name>
        <value>604800</value>
    </property>
</configuration>
```

7. 配置slaves

```
hd01
hd02
hd03
```

8. 分发安装包到hd02，hd03

```
scp -r hadoop260/ root@hd02:/opt/soft/
scp -r hadoop260/ root@hd03:/opt/soft/
```

9. 为3台节点配置hadoop环境变量(vi /etc/profile)

```
#hadoop
export HADOOP_HOME=/opt/soft/hadoop260
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

4. 启动集群
    1. 启动zookeeper(3个都启动)

    ```
    zkServer.sh start
    ```

    2. 启动JournalNode(3个都启动)

    ```
    hadoop-deamon.sh start journode
    ```

    3. 格式化namenode(只在hd01主机上)

    ```
    hdfs namenode -format
    ```

    4. 将hd01上的Namenode的元数据复制到hd02相同位置

    ```
    scp -r /opt/soft/hadoop260/dfs/namenode_data/current/
    root@hd02:/opt/soft/hadoop260/dfs/namenode_data
    ```

    5. 在hd01或hd02格式化故障转移控制器zkfc

    ```
    hdfs zkfc -formatZK
    ```

    6. 在hd01上启动dfs服务

    ```
    start-dfs.sh
    ```

    7. 在hd03上启动yarn服务

    ```
    start-yarn.sh
    ```

    8. 在hd01上启动history服务器

    ```
    mr-jobhistory-daemon.sh start historyserver
    ```

    9. 在hd02上启动resourcemanager服务

    ```
    yarn-deamon.sh start resourcemanager
    ```

检查集群情况

1. jps 上面服务不能缺少
2. 查看状态

```
# 在hd01上查看服务状态
hdfs haadmin -getServiceState nn1   #active
hdfs haadmin -getServiceState nn2   #standby
# 在hd03上查看resourcemanager状态
yarn rmadmin -getServiceState rm1   #standby
yarn rmadmin -getServiceState rm2   #active
```

3. 检查主备切换

```
# kill 掉Namenode主节点 查看Namenode standby节点状态
kill -9 namenode主节点进程
# 恢复后重新加入
hadoop-deamon.sh start namenode #启动后也只是standby节点
```

集群二次启动

```
#在hd01上启动dfs
start-dfs.sh
#在hd03上启动yarn
start-yarn.sh
#在hd02上启动resourcemanager
yarn-daemon.sh start resourcemanager
```