

21-namenode高可用2 (ha) (热备) -配置-及启动

-----成都尚学堂-mr-zeng-----

集群进程的安排

Ps->简单版-实际开发中基本一个进程一个节点

主机名	IP	NameNode	ZKFC	DataNode	Yarn	Zookeeper	JournalNode
mast1	192.168.177.131	是	是	是	否	是	是
mast2	192.168.177.132	是	是	是	否	是	是
mast3	192.168.177.133	否	否	是	是	是	是

集群安装配置

ps—》下面有很多配置文件-》为了方便这里可以使用xftp(文件夹操作使用xshell, 小心权限问题, 名字问题)---》实际开发多用配置管理平台-cdh-统一管理

1) 各节点配置好【网络和域名映射】

2) 关闭防火墙

3) ssh免登录

4) 安装jdk

5) zookeeper集群搭建

a) xftp复制zookeeper-3.4.10.tar.gz到 -》 mast1的/root

b) xshell登录到mast1, 进行解压

```
tar -zxvf /root/ zookeeper-3.4.10.tar.gz -C /usr/zookeeper
```

c) 配置文件修改为如下

配置文件路径 /usr/zookeeper/zookeeper-3.4.10/conf

x) 先把文件名zoo_sample.cfg改为zoo.cfg

x) 修改zoo.cfg内容参考如下 (最少需要更改机器对应的域名 (绿色))

```
#基本时间单元2秒
tickTime=2000
#客户端连接端口
clientPort=2181
#客户端初始连接服务器的最大时间
initLimit=5
#follower连接leader的最大时间
syncLimit=4
#存储数据位置
dataDir=/zookeeper/data
dataLogDir=/zookeeper/datalog
#server.X 这个数字就是对应 data/myid中的数字。你在3个server的myid文件中分别写入了1, 2, 3,
#那么每个server中的zoo.cfg都写server.1,server.2,server.3就OK了。因为在同一台机器上,
#后面连着的2个端口3个server都不要一样, 否则端口冲突, 其中第一个端口用来集群成员的信息交换,
#第二个端口是在leader挂掉时专门用来进行选举leader所用。
server.1=mast1:2888:3888
server.2=mast2:2888:3888
server.3=mast3:2888:3888
```

d) 使用scp命令把mast1的/usr/zookeeper复制给其他节点mast2和mast3

e) 在每一个节点/zookeeper/data建立文件myid

ps-》里面存储一个数字 (用来标识当前服务器) 该数字为zoo.cfg里【绿色域名】前的数字

f) 分别启动每一个zkServer (自动选出一个leader总统)

```
cd /usr/zookeeper/zookeeper-3.4.10/bin
./zkServer.sh start
```

g) 检查集群是否正常

```
./zkServer.sh status
```

如下图-显示两个follower 一个leader则搭建成功

```
1 Thinkpad_node21 + 1 Thinkpad_node23 +
[root@node21 data]# jps
2064 Jps
[root@node21 data]# zkServ
ZooKeeper JMX enabled by d
Using config: /usr/zookeep
Starting zookeeper ... STA
[root@node21 data]# zkServ
ZooKeeper JMX enabled by d
Using config: /usr/zookeep
Mode: follower
[root@node21 data]#

Using config: /usr/zookeeper/z
Starting zookeeper ... STARTED
[root@node23 data]# zkServer.s
ZooKeeper JMX enabled by defau
Using config: /usr/zookeeper/z
Mode: leader
[root@node23 data]#

1 Thinkpad_node22 +
[root@node22 data]# jps
5687 Jps
[root@node22 data]# zkServer.s
ZooKeeper JMX enabled by defau
Using config: /usr/zookeeper/z
Starting zookeeper ... STARTED
[root@node22 data]# zkServer.s
ZooKeeper JMX enabled by defau
Using config: /usr/zookeeper/z
Mode: follower
[root@node22 data]#
```

6) hadoop集群搭建

(hdfs) ps—》 Namenode , datanode , JournalNode , FailoverController-注意这里不需要secondaryNamenode

(yarn) ps—》 resourceManager , nodeManager

如果-需要使用yarn和mapreduce (自行参考之前资料-》配置mapred-site.xml和yarn-site.xml)

hdfs集群搭建 (ha高可用的)

-----先在mast1配置下面信息-----

Core.xml—》 hdfs核心信息

```
<configuration>
<!-- 指定hdfs的nameservice为ns -->
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://ns</value>
</property>
<!-- 指定hadoop数据临时存放目录-->
<property>
  <name>hadoop.tmp.dir</name>
  <value>/hadoop2.72_data/tmp</value>
</property>
<!-- 指定zookeeper地址-->
<property>
  <name>ha.zookeeper.quorum</name>
  <value>mast1:2181,mast2:2181,mast3:2181</value>
</property>
</configuration>
```

Hdfs-site.xml—》 namenode , JournalNode信息zkfc (不用配置-自动绑定namenode)

```
<configuration>
<!-- 指定hdfs的nameservice为ns , 需要和core-site.xml中的保持一致 -->
<property>
  <name>dfs.nameservices</name>
  <value>ns</value>
</property>
<!-- ns下面有两个NameNode , 分别是nn1 , nn2 -->
<property>
  <name>dfs.ha.namenodes.ns</name>
  <value>nn1,nn2</value>
</property>
<!-- nn1的RPC通信地址 -->
<property>
  <name>dfs.namenode.rpc-address.ns.nn1</name>
  <value>mast1:9000</value>
</property>
<!-- nn1的http通信地址 -->
<property>
  <name>dfs.namenode.http-address.ns.nn1</name>
  <value>mast1:50070</value>
</property>
<!-- nn2的RPC通信地址 -->
<property>
  <name>dfs.namenode.rpc-address.ns.nn2</name>
  <value>mast2:9000</value>
</property>
```

```

<!-- nn2的http通信地址 -->
<property>
  <name>dfs.namenode.http-address.ns.nn2</name>
  <value>mast2:50070</value>
</property>
<!-- 指定NameNode元数据-JournalNode的（服务节点和端口）-->
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>qjournal://mast1:8485;mast2:8485;mast3:8485/ns</value>
</property>
<!-- 指定JournalNode在本地磁盘存放数据的位置 -->
<property>
  <name>dfs.journalnode.edits.dir</name>
  <value>/hadoop2.72_jn_name/tmp</value>
</property>
<!-- 开启NameNode故障时自动切换 -->
<property>
  <name>dfs.ha.automatic-failover.enabled</name>
  <value>true</value>
</property>
<!-- 配置失败自动切换实现方式 -->
<property>
  <name>dfs.client.failover.proxy.provider.ns</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>

<!-- 配置隔离机制 -->
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence</value>
</property>
<!-- 使用隔离机制时需要ssh免登陆 -->
<property>
  <name>dfs.ha.fencing.ssh.private-key-files</name>
  <value>/root/.ssh/id_dsa</value>
</property>

<!-- 在NN和DN上开启WebHDFS (REST API)功能(通过http访问hdfs),不是必须 -->
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
</configuration>

```

Slaves—》Datanode (如果使用yarn-nodemanager也使用该配置)

```

mast1
mast2
mast3

```

修改JAVA_HOME

分别在文件**hadoop-env.sh**中添加JAVA_HOME配置

```
#export JAVA_HOME=${JAVA_HOME} --原来
```

```
export JAVA_HOME=/usr/java/jdk1.7.0_79
```

ps》虽然默认配置了\$(JAVA_HOME)的环境变量，但是hadoop启动时，会提示找不到，没有办法，指定绝对路径，这个是必须的。

```
#export JAVA_HOME=${JAVA_HOME} --原来
export JAVA_HOME=/usr/local/java/jdk1.7.0_67
```

虽然默认配置了\$(JAVA_HOME)的环境变量，但是hadoop启动时，会提示找不到，没有办法，指定绝对路径，这个是必须的。

配置hadoop的环境变量，参考如下配置

```
vi /etc/profile
```

追加下面内容

```
done
unset i
unset -f pathmunge
export JAVA_HOME=/usr/java/jdk1.7.0_79
export HADOOP_HOME=/usr/hadoop/hadoop-2.7.2
export ZOOKEEPER_HOME=/usr/zookeeper/zookeeper-3.4.10
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$ZOOKEEPER_HOME/bin
```

java的路径
hadoop路径
zookeeper路径

把java,hadoop,zookeeper里要使用的命令程序配置到path, 可以方便直接使用对应命令

复制mast1的上面的配置到mast2和mast3

如下方式

- 使用命令scp完成
- 或xftp的copy
- 最简单的方式 (先配置一个节点, 再克隆该节点!! 新节点需要配置网络和映射, 所有节点防火墙关闭)

7) hadoop和zookeeper集群启动

ps-》(初次-需要一些初始化操作)

ps-》建议每启动一个jps查看检查一下

-----zookeeper集群-----

a) 启动zookeeper集群 (由于 (hadoop里) zkfc工作的时候需要使用zookeeper-必须先启动zookeeper)

- 启动分别在mast1、mast2、mast3上执行如下命令启动zookeeper集群;
zkServer.sh start
- 验证启动
zkServer.sh status 查看是否有leader

-----hadoop集群-----在mast1操作即可-----

b) 格式化zkfc, 让在zookeeper中生成ha要使用的数据节点

```
hdfs zkfc -formatZK
```

c) 启动journalnode集群 (namenode工作的时候需要使用元数据-必须先启动journalnode)

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

d) 格式化hdfs

```
hadoop namenode -format
```

e) 启动NameNode

- (mast1操作) 首先在mast1上启动active节点, 在mast1上执行如下命令
sbin/hadoop-daemon.sh start namenode
- (mast2操作) 在mast2上同步namenode的数据, 同时启动mast2的standby的namenode, 命令如下
#把NameNode的数据同步到mast2上
hdfs namenode -bootstrapStandby
#启动mast2上的namenode作为standby
sbin/hadoop-daemon.sh start namenode

f) 启动datanode

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

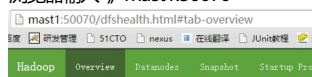
g) 启动ZKFC

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

-----测试HA的高可用性-----

a) 访问mast1的namenode

浏览器输入-》mast1:50070



Overview 'Mast1:9000' (active)

b) 访问mast2的namenode

浏览器输入-》mast2:50070

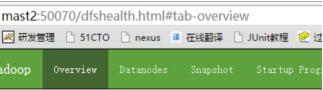


Overview 'Mast2:9000' (standby)

c) 关闭mast1的namenode

- x) 使用ps查看mast1的进程d
- x) 使用kill -9 进程d 杀死mast1的namenode

d) 再次访问mast2的namenode



Overview 'Mast2:9000' (active)

-----再次启动-----

- a) 启动zookeeper
zkServer.sh start
- b) 启动hdfs
start-dfs.sh