Hadoop ha集群搭建

# 环境准备

1 虚拟机安装

Linux采用CentOS-7-x86\_64-Minimal-1810.iso，安装过后进入系统设置以下：

#用root将hadoop加入sudo组

vi /etc/sudoers

#关闭防火墙

sudo systemctl stop iptables

sudo systemctl disable iptables

#关闭图形网络管理

sudo systemctl stop NetworkManager

sudo systemctl disable NetworkManager

#编辑网卡属性，加入以下配置

sudo vi /etc/sysconfig/network-scripts/ifcfg-ens33

TYPE=Ethernet

BOOTPROTO=static

NAME=ens33

DEVICE=ens33

ONBOOT=yes

IPADDR=192.168.47.131

GATEWAY=192.168.47.2

DNS1=192.168.47.2

#重启网卡

sudo systemctl restart network

sudo yum install net-tools -y

#psmisc包含fuser，killall，pstree三个程序， centos7最小化安装，默认是不安装psmics。ha会用到fuser

sudo yum install psmisc -y

#关闭IPv6

sudo vi /etc/modprobe.d/dist.conf    #在该文件末尾加上以下语句

alias net-pf-10 off

alias ipv6 off    #保存后重启即可

#调整文件的最大打开数

ulimit -a    #查看文件的最大打开数 ulimit -n也可以

sudo vi /etc/security/limits.conf    #添加以下配置

\* soft nofile 65535

\* hard nofile 65535

\* soft nproc 65535

\* hard nproc 65535

#用 swap 交换区,修改内核参数

sudo vi /etc/sysctl.conf    #添加配置

net.core.somaxconn = 32768 # 表示物理内存使用到 90%（100-10=90）的时候才使

#关闭sellinux防火墙

getenforce    #查看sellinux防火墙安全策略

setenforce    #临时设置sellinux的安全策略

/usr/bin/sellinux    #查看详细信息

sudo vi /etc/sellinux/config    #修改配置

SELINUX=disabled

#配置hosts

sudo vi /etc/hosts

192.168.47.131 node1

192.168.47.132 node2

192.168.47.133 node3

#配置hostname

sudo vi /etc/hostname

node1

# 集群部署

## 1 免密登陆

#每台机器都要执行

sudo systemctl enable sshd

sudo systemctl start sshd

ssh-keygen -t rsa -P ‘’ -f ~/.ssh/id\_rsa

ssh-copy-id node1

ssh-copy-id node2

ssh-copy-id node3

## 2 上传安装包

Linux远程连接工具采用secureCRT，当前session下按快捷键alt+p出现sftp界面，拖拉文件上传。

## 3 解压安装包

tar zxvf jdk-8u191-linux-x64.tar.gz -C ~/app/

tar zxvf zookeeper-3.4.9.tar.gz -C ~/app/

tar zxvf hadoop-2.7.2.tar.gz -C ~/app/

## 4 配置jdk

#查看是否安装其它版本jdk

sudo rpm -qa | grep jdk

#若有可用如下命令删除

sudo rpm -e –nodeps 软件包名

mv app/jdk1.8.0\_191 app/java

#将Java相关加入环境变量

sudo vi /etc/profile

export JAVA\_HOME=/home/hadoop/app/java

export CLASS\_PATH=/home/hadoop/app/java/jre/lib:/home/hadoop/app/java/lib

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

#使配置生效

source /etc/profile

#查看是否安装成功

java -version

## 5 zk部署

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial

# synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/home/hadoop/app/zookeeper/data

# the port at which the clients will connect

clientPort=2181

# the maximum number of client connections.

# increase this if you need to handle more clients

#maxClientCnxns=60

mv app/zookeeper-3.4.9 app/zookeeper

cd app/zookeeper/conf

mv zoo\_sample.cfg zoo.cfg

#zookeeper配置，

vi zoo.cfg

#配置结束

cd ..

mkdir data

#分发到其他机器并根据配置文件修改myid

echo 1 > data/myid

bin/zkServer.sh start

scp -r ../zookeeper hadoop@node2:/home/hadoop/app

--echo 2 > data/myid

--bin/zkServer.sh start

scp -r ../zookeeper hadoop@node3:/home/hadoop/app

--echo 3 > data/myid

--bin/zkServer.sh start

#

# Be sure to read the maintenance section of the

# administrator guide before turning on autopurge.

#

# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# The number of snapshots to retain in dataDir

#autopurge.snapRetainCount=3

# Purge task interval in hours

# Set to "0" to disable auto purge feature

server.1=node1:2888:3888

server.2=node2:2888:3888

server.3=node3:2888:3888

## 6 hadoop ha部署

### env配置

mv app/hadoop-2.7.2 app/hadoop

cd app/hadoop

mkdir data

cd etc/hadoop

#环境变量导入省略

vi hadoop-env.sh

vi yarn-env.sh

#加入java配置

export JAVA\_HOME=/home/hadoop/app/java

#结束

### core-site.xml

<configuration>

    <property>

  <name>fs.defaultFS</name>

        <value>hdfs://ns1</value>

    </property>

    <property>

        <name>hadoop.tmp.dir</name>

        <value>/home/hadoop/app/hadoop/data/tmp</value>

    </property>

    <property>

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

        <value>node1:2181,node2:2181,node3:2181</value>

    </property>

</configuration>

### hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.permissions.enabled</name>

<value>false</value>

</property>

<property>

<name>dfs.nameservices</name>

<value>ns1</value>

</property>

<property>

<name>dfs.blocksize</name>

<value>134217728</value>

</property>

<property>

<name>dfs.ha.namenodes.ns1</name>

<value>nn1,nn2</value>

</property>

<!-- nn1的RPC通信地址，nn1所在地址 -->

<property>

<name>dfs.namenode.rpc-address.ns1.nn1</name>

<value>node1:8020</value>

</property>

<!-- nn1的http通信地址，外部访问地址 -->

<property>

<name>dfs.namenode.http-address.ns1.nn1</name>

<value>node1:50070</value>

</property>

<!-- nn2的RPC通信地址，nn2所在地址 -->

<property>

<name>dfs.namenode.rpc-address.ns1.nn2</name>

<value>node2:8020</value>

</property>

<!-- nn2的http通信地址，外部访问地址 -->

<property>

<name>dfs.namenode.http-address.ns1.nn2</name>

<value>node2:50070</value>

</property>

<!-- 指定NameNode的元数据在JournalNode日志上的存放位置(一般和zookeeper部署在一起) -->

<property>

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

<value>qjournal://node1:8485;node2:8485;node3:8485/ns1</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>/home/hadoop/app/hadoop/data/journal</value>

</property>

mapred-site.xml

<!--客户端通过代理访问namenode，访问文件系统，HDFS 客户端与Active 节点通信的Java 类，使用其确定Active 节点是否活跃 -->

<property>

<name>dfs.client.failover.proxy.provider.ns1</name>

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

</property>

<!--这是配置自动切换的方法，有多种使用方法，具体可以看官网，在文末会给地址，这里是远程登录杀死的方法 -->

<property>

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

<value>sshfence</value>

</property>

<!-- 这个是使用sshfence隔离机制时才需要配置ssh免登陆 -->

<property>

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

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

</property>

<!-- 配置sshfence隔离机制超时时间，这个属性同上，如果你是用脚本的方法切换，这个应该是可以不配置的 -->

<property>

<name>dfs.ha.fencing.ssh.connect-timeout</name>

<value>30000</value>

</property>

<!-- 这个是开启自动故障转移，如果你没有自动故障转移，这个可以先不配 -->

<property>

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

<value>true</value>

</property>

</configuration>

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>node1:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node1:19888</value>

</property>

</configuration>

### yarn-site.xml

<configuration>

    <property>

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

<value>mapreduce\_shuffle</value>

    </property>

    <!-- Site specific YARN configuration properties -->

    <!--启用resourcemanager ha-->

    <!--是否开启RM ha，默认是开启的-->

    <property>

     <name>yarn.resourcemanager.ha.enabled</name>

     <value>true</value>

    </property>

    <!--声明两台resourcemanager的地址-->

    <property>

     <name>yarn.resourcemanager.cluster-id</name>

     <value>rmcluster</value>

    </property>

    <property>

     <name>yarn.resourcemanager.ha.rm-ids</name>

     <value>rm1,rm2</value>

    </property>

    <property>

     <name>yarn.resourcemanager.hostname.rm1</name>

     <value>node1</value>

    </property>

    <property>

     <name>yarn.resourcemanager.hostname.rm2</name>

     <value>node2</value>

    </property>

    <!--指定zookeeper集群的地址-->

    <property>

     <name>yarn.resourcemanager.zk-address</name>

        <value>node1:2181,node2:2181,node3:2181</value>

    </property>

    <!--启用自动恢复，当任务进行一半，rm坏掉，就要启动自动恢复，默认是false-->

    <property>

     <name>yarn.resourcemanager.recovery.enabled</name>

     <value>true</value>

    </property>

    <!--指定resourcemanager的状态信息存储在zookeeper集群，默认是存放在FileSystem里面。-->

    <property>

     <name>yarn.resourcemanager.store.class</name>

     <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>

    </property>

<property>

<name>yarn.log-aggregation-enable</name>

<value>true</value>

</property>

<property>

<property>

<name>yarn.log.server.url</name>

<value>http://node1:19888/jobhistory/logs</value>

</property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

</property>

</configuration>

### 配置slaves

node2

node3

### 分发hadoop

scp -r ~/app/hadoop hadoop@node2:/home/hadoop/app

scp -r ~/app/hadoop hadoop@node3:/home/hadoop/app

### 集群初始化

1.启动zk

2.启动journalnode:

hadoop-daemons.sh start journalnode

3.格式化zkfc--让在zookeeper中生成ha节点

hdfs zkfc –formatZK

4.格式化hdfs

hdfs namenode –format

5.启动NameNode（node1）

hadoop-daemon start namenode

6.standby同步namenode的数据，并启动(node2)

hdfs namenode –bootstrapStandby

7.启动启动datanode

hadoop-daemons.sh start datanode

8.启动yarn

sbin/start-yarn.sh

9.启动zkfc

hadoop-daemons.sh start zkfc

10.访问http://node1:50070验证