搭建开源HADOOP集群

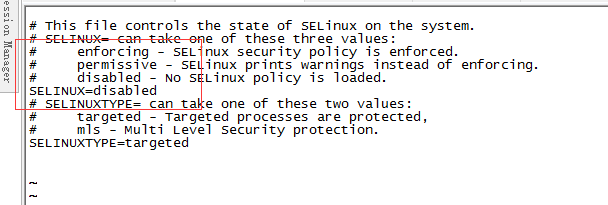
一．搭建前准备

1.三台节点在root下，关闭iptables和selinux

service iptables stop //当前关闭防火墙

chkconfig iptables off //永久关闭防火墙

vi /etc/sysconfig/selinux //调整当前系统的selinux状态为disabled的



2.配置hosts文件

vi /etc/hosts[三台都需要添加如下内容]，添加主机ip和对应的主机名，ip和主机名中间用空格分开，一行一个。

172.18.18.114  s114

       172.18.18.115  s115

       172.18.18.116  s116

将hosts文件拷贝到s115和s116，执行以下命令

scp -r /etc/hosts s115:/etc/hosts //主机名请根据情况修改

scp -r /etc/hosts  s116:/etc/hosts //主机名请根据情况修改

3.配置ssh无密钥登录，以下命令在root用户下执行

s114执行以下命令：

        ssh-keygen -t rsa

        ssh-copy-id s114

        ssh-copy-id s115

        ssh-copy-id s116

s115执行以下命令：

         ssh-keygen -t rsa

        ssh-copy-id s114

        ssh-copy-id s115

        ssh-copy-id s116

s116执行以下命令：

      ssh-keygen -t rsa

        ssh-copy-id s114

        ssh-copy-id s115

        ssh-copy-id s116

二．安装jdk

1.下载jdk安装包

2.上传至s114节点的/usr/local目录下

3.进入/usr/local,解压安装包

cd /usr/local

tar -zxvf jdk-8u144-linux-x64.tar.gz -C /usr/local //解压

4.配置环境变量

vim /etc/profile

export JAVA\_HOME=/usr/local/jdk //在文件末尾添加内容

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

Source /etc/profile //更新

5.拷贝到s115.s116节点

scp -r /usr/local/jdk s115:/usr/local

scp -r /usr/local/jdk s116:/usr/local

scp -r /etc/profile s115:/etc/profile //拷贝文件到其他机器

在s115上执行source /etc/profile

scp -r /etc/profile s115:/etc/profile

在s116上执行source /etc/profile

三．安装配置zookeeper

1.下载zookeeper-3.4.6.tar.gz，上传至/usr/local目录，cd /usr/local

解压安装，tar -zxvf zookeeper-3.4.6.tar.gz

2.chmod -R 755 /usr/local/zookeeper-3.4.6

3.修改 vim /usr/local/zookeeper-3.4.6/conf/zoo.cfg

dataDir=/usr/local/zookeeper-3.4.6/data //修改dataDir

server.1=172.18.18.114:2888:3888 //在文件末尾添加以下内容，ip请根据情况修改

server.2=172.18.18.115:2888:3888

server.3=172.18.18.116:2888:3888

4.创建zookeeper的data目录 mkdir /usr/local/zookeeper-3.4.6/data

5.cd /usr/local/zookeeper-3.4.6/data

vim myid ，按a进去编辑模式，写入内容1，按esc退出编辑模式，:wq保存退出。

6.分发至s115和s116

scp -r /usr/local/zookeeper-3.4.6/ s115:/usr/local

scp -r /usr/local/zookeeper-3.4.6/ s116:/usr/local

7.分别在s115和s116上修改 vim /usr/local/zookeeper-3.4.6/data/myid，s115写入内容为2，s116写入内容为3

8.分别在s114，s115，s116上启动zookeeper

/usr/local/zookeeper-3.4.6/bin/zkServer.sh start

9.jps查看有以下进程

QuorumPeerMain

四．安装配置分布式高可用hadoop集群

1.下载hadoop-2.7.2.tar.gz，上传至/usr/local目录。

2.进入目录，cd /usr/local/

3.解压tar -zxvf hadoop-2.7.2.tar.gz

4.chmod -R 755 hadoop-2.7.2

5.修改vim /usr/local/hadoop-2.7.2/etc/hadoop/hadoop-env.sh，修改vim /usr/local/hadoop-2.7.2/etc/hadoop/yarn-env.sh

添加JAVA\_HOME

export JAVA\_HOME=/usr/local/jdk

修改vim /usr/local/hadoop-2.7.2/etc/hadoop/slaves，添加从节点，一行一个,分别为s115和s116主机名

s115

s116

6.修改 vim /usr/local/hadoop-2.7.2/etc/hadoop/core-site.xml

<!-- 指定hdfs的nameservice为ns1 -->

<property>

<name>fs.defaultFS</name>

<value>hdfs://hzgc/</value>

</property>

<!-- 指定hadoop临时目录 -->

<property>

<name>hadoop.tmp.dir</name>

<value>/usr/local/hadoop-2.7.2/tmp</value>

</property>

<!-- 指定zookeeper地址 -->

<property>

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

<value>s114:2181,s115:2181,s116:2181</value>

</property>

7.修改 vim /usr/local/hadoop-2.7.2/etc/hadoop/hdfs-site.xml

<!--指定hdfs的nameservice为hzgc，需要和core-site.xml中的保持一致 -->

<property>

<name>dfs.nameservices</name>

<value>hzgc</value>

</property>

<!-- dksou下面有两个NameNode，分别是nn1，nn2 -->

<property>

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

<value>nn1,nn2</value>

</property>

<!-- nn1的RPC通信地址 -->

<property>

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

<value>s114:9000</value>

</property>

<!-- nn1的http通信地址 -->

<property>

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

<value>s114:50070</value>

</property>

<!-- nn2的RPC通信地址 -->

<property>

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

<value>s115:9000</value>

</property>

<!-- nn2的http通信地址 -->

<property>

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

<value>s115:50070</value>

</property>

<!-- 指定NameNode的元数据在JournalNode上的存放位置 -->

<property>

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

<value>qjournal://s114:8485;s115:8485;s116:8485/hzgc</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>/usr/local/hadoop-2.7.2/data/tmp/journal</value>

</property>

<!-- 开启NameNode失败自动切换 -->

<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.hzgc</name>

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

</property>

<!-- 配置隔离机制方法，多个机制用换行分割，即每个机制暂用一行-->

<property>

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

<value>

sshfence

shell(/bin/true)

</value>

</property>

<!-- 使用sshfence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\_rsa</value>

</property>

<!-- 配置sshfence隔离机制超时时间 -->

<property>

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

<value>30000</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

8.修改vim /usr/local/hadoop-2.7.2/etc/hadoop/mapred-site.xml，

<!-- 指定mr框架为yarn方式 -->

<property>

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

<value>yarn</value>

</property>

9.修改 vim /usr/local/hadoop-2.7.2/etc/hadoop/yarn-site.xml

<!-- 开启RM高可用 -->

<property>

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

<value>true</value>

</property>

<!-- 指定RM的cluster id -->

<property>

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

<value>yrc</value>

</property>

<!-- 指定RM的名字 -->

<property>

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

<value>rm1,rm2</value>

</property>

<!-- 分别指定RM的地址 -->

<property>

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

<value>s114</value>

</property>

<property>

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

<value>s115</value>

</property>

<!-- 指定zk集群地址 -->

<property>

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

<value>s114:2181,s115:2181,s116:2181</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

10.将hadoop分发到另外两台服务器

scp -r /usr/local/hadoop-2.7.2 s115:/usr/local

chmod -R 755 /usr/local/hadoop-2.7.2 //在s115上执行

scp -r /usr/local/hadoop-2.7.2 s116:/usr/local

chmod -R 755 /usr/local/hadoop-2.7.2 //在s116上执行

11.每个节点都启动zookeeper（若zookeeper服务未启动）

/usr/local/zookeeper-3.4.6/bin/zkServer.sh start

12.只在第一台s114格式化zkfc

/usr/local/hadoop-2.7.2/bin/hdfs zkfc -formatZK -force

13.只在第一台s114启动zkfc

/usr/local/hadoop-2.7.2/sbin/hadoop-daemon.sh start zkfc

14.每个节点都启动journalnode

/usr/local/hadoop-2.7.2/sbin/hadoop-daemon.sh start journalnode

15.启动hadoop集群

（1）确保zookeeper集群启动成功

（2）在nn1上，格式化namenode

/usr/local/hadoop-2.7.2/bin/hdfs namenode -format -force

（3）启动hdfs（在nn1上执行）

/usr/local/hadoop-2.7.2/sbin/start-dfs.sh

（4）在nn2上，同步nn1的元数据信息

/usr/local/hadoop-2.7.2/bin/hdfs namenode -bootstrapStandby

（5）启动yarn（在nn1上执行）

/usr/local/hadoop-2.7.2/sbin/start-yarn.sh

（6）启动hdfs，yarn的ha

直接在nn2上执行/usr/local/hadoop-2.7.2/sbin/hadoop-daemon.sh start namenode

直接在nn2上执行/usr/local/hadoop-2.7.2/sbin/yarn-daemon.sh start resourcemanager

（7）启动hadoop历史记录服务（在nn1或nn2任选一台起，或者都起）

/usr/local/hadoop-2.7.2/sbin/mr-jobhistory-daemon.sh start historyserver

1. 验证

在s114上jps查看

Namenode，ResourceManager，DFSZKFailoverController，JournalNode，HistoryServer

在s115上jps查看

Namenode，ResourceManager，datanode，JournalNode，HistoryServer

在s116上jps查看

datanode，NodeManager，JournalNode

五．安装配置hbase集群

1. 下载hbase-1.2.0.tar.gz，，上传至/usr/local目录，cd /usr/local

解压安装，tar -zxvf hbase-1.2.0.tar.gz

1. chmod -R 755 hbase-1.2.0

3.修改 vim hbase-env.sh

export JAVA\_HOME=/opt/tool/jdk

export HBASE\_MANAGES\_ZK=false

4.修改 vim hbase-site.xml

<property>

<name>hbase.rootdir</name>

<value>hdfs://hzgc/hbase</value>

</property>

<property>

<name>hbase.master.port</name>

<value>60000</value>

</property>

<property>

<name>hbase.regionserver.port</name>

<value>16020</value>

</property>

<property>

<name>hbase.regionserver.info.port</name>

<value>16030</value>

</property>

<property>

<name>hbase.master.info.port</name>

<value>60010</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.property.clientPort</name>

<value>2181</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>s114:2181,s115:2181,s116:2181</value>

</property>

<property>

<name>hbase.tmp.dir</name>

<value>/usr/local/hbase-1.2.0/tmp</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/usr/local/zookeeper-3.4.6/data</value>

</property>

<property>

<name>hbase.master.maxclockskew</name>

<value>300000</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

5.修改/usr/local/hbase-1.2.0/conf/regionservers，添加从节点地址，一行一个，内容为s115和s116

s115

s116

6.将拷贝hadoop相关配置到hbase的conf目录下

cp -r /usr/local/hadoop-2.7.2/etc/hadoop/core-site.xml,/usr/local/hadoop-2.7.2/etc/hadoop/hdfs-site.xml

7.将/usr/local/hbase-1.2.0/ 分发至其他节点

scp -r /usr/local/hbase-1.2.0/ s115:/usr/local

scp -r /usr/local/hbase-1.2.0/ s116:/usr/local

8.在每个节点启动zookeeper（如果没起的话）

9.在s114上启动hbase集群

/usr/local/hbase-1.2.0/bin/start-hbase.sh

1. 启动hbase ha

直接在s115上启动

/usr/local/hbase-1.2.0/bin/hbase-daemon.sh start hmaster

1. 验证

在s114上jps查看

Hmaster

在s115上jps查看

Hmaster

HRegionServer

在s116上jps查看

HRegionServer

六．安装配置spark集群

1. 下载scala-2.11.8.tgz，上传至目录/usr/local，cd /usr/local

解压tar -zxvf scala-2.11.8.tgz

2.chmod -R 755 /usr/local/scala-2.11.8，分发至其他节点

scp -r /usr/local/scala-2.11.8 s115:/usr/local

scp -r /usr/local/scala-2.11.8 s116:/usr/local

3.每台节点都配置系统变量

vim /etc/profile //在文件最后加上以下内容

export SCALA\_HOME=/usr/local/scala-2.11.8

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

source /etc/profile //更新

4.验证scala

scala -version //出现scala的信息

5.编译spark源码

（1）下载spark-2.2.0源码，下载并配置maven环境变量

vim /etc/profile //增加以下内容

export MAVEN\_HOME=/usr/local/apache-maven-3.5.0

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

（2）使用使用make-distribution编译

cd /usr/local/spark-2.2.0/dev

修改vim make-distribution

VERSION=2.2.0

SCALA\_VERSION=2.11.8

SPARK\_HADOOP\_VERSION=2.7.2

SPARK\_HIVE=1

（3）spark-2.2.0/dev/make-distribution.sh --tgz \

-Phadoop-2.7.2 -Dhadoop.version=2.7.2 \

-Pyarn \

-Phive -Phive-thriftserver

(4) 在/usr/local/spark-2.2.0/target的下可以找spark-2.2.0-bin-2.7.2.tgz

1. 安装spark-2.2.0-bin-2.7.2.tgz，上传至目录/usr/local，cd /usr/local

解压tar -zxvf spark-2.2.0-bin-2.7.2.tgz,chmod -R 755 park-2.2.0-bin-2.7.2

6.修改 vim /usr/local/spark-2.2.0-bin-2.7.2/conf/spark-env.sh

export SPARK\_DAEMON\_JAVA\_OPTS="-Dspark.deploy.recoveryMode=ZOOKEEPER -Dspark.deploy.zookeeper.url=s114:2181,s115:2181,s116:2181 -Dspark.deploy.zookeeper.dir=/spark"

export JAVA\_HOME=/opt/tool/jdk

export HADOOP\_HOME=/usr/local/hadoop-2.7.2

export HADOOP\_CONF\_DIR=${HADOOP\_HOME}/etc/hadoop

export SCALA\_HOME=/usr/local/scala-2.11.8

export SPARK\_MASTER\_IP=172.18.18.114 #

export SPARK\_MASTER\_PORT=7077 #

export SPARK\_MASTER\_WEBUI\_PORT=8080 #

export SPARK\_LOCAL\_DIR=/opt/client/Spark/tmp #

export SPARK\_HISTORY\_OPTS="-Dspark.history.ui.port=7777 -Dspark.history.fs.logDirectory=hdfs://hzgc/sparkJobHistory" #

export SPARK\_WORKER\_MEMORY=4g

export SPARK\_EXECUTOR\_MEMORY=2g

export SPARK\_DRIVER\_MEMORY=1g

export SPARK\_WORKER\_CORES=4

export SPARK\_CLASSPATH=/usr/local/spark-2.2.0-bin-hadoop2.7.2/lib/mysql-connector-java.jar

export SPARK\_CLASSPATH=$SPARK\_CLASSPATH:$CLASSPATH

7.修改 vim /usr/local/spark-2.2.0-bin-2.7.2/conf/spark-defaults.conf

spark.eventLog.enabled true

spark.eventLog.dir=hdfs://hzgc/sparkJobHistory

#spark.eventLog.dir /usr/local/spark-2.2.0-bin-hadoop2.7.2/log/historyserverforSpark

#spark.yarn.historyServer.address s114:18080

#spark.history.fs.logDirectory /usr/local/spark-2.2.0-bin-hadoop2.7.2/log

8.修改/usr/local/spark-2.2.0-bin-2.7.2/conf/slaves，添加从节点地址，一行一个

s115

s116

修改过后将/usr/local/spark-2.2.0-bin-2.7.2分发至其他节点

scp -r /usr/local/spark-2.2.0-bin-2.7.2 s115:/usr/local/

scp -r /usr/local/spark-2.2.0-bin-2.7.2 s116:/usr/local/

9.启动spark集群

确保hdfs和yarn启动成功

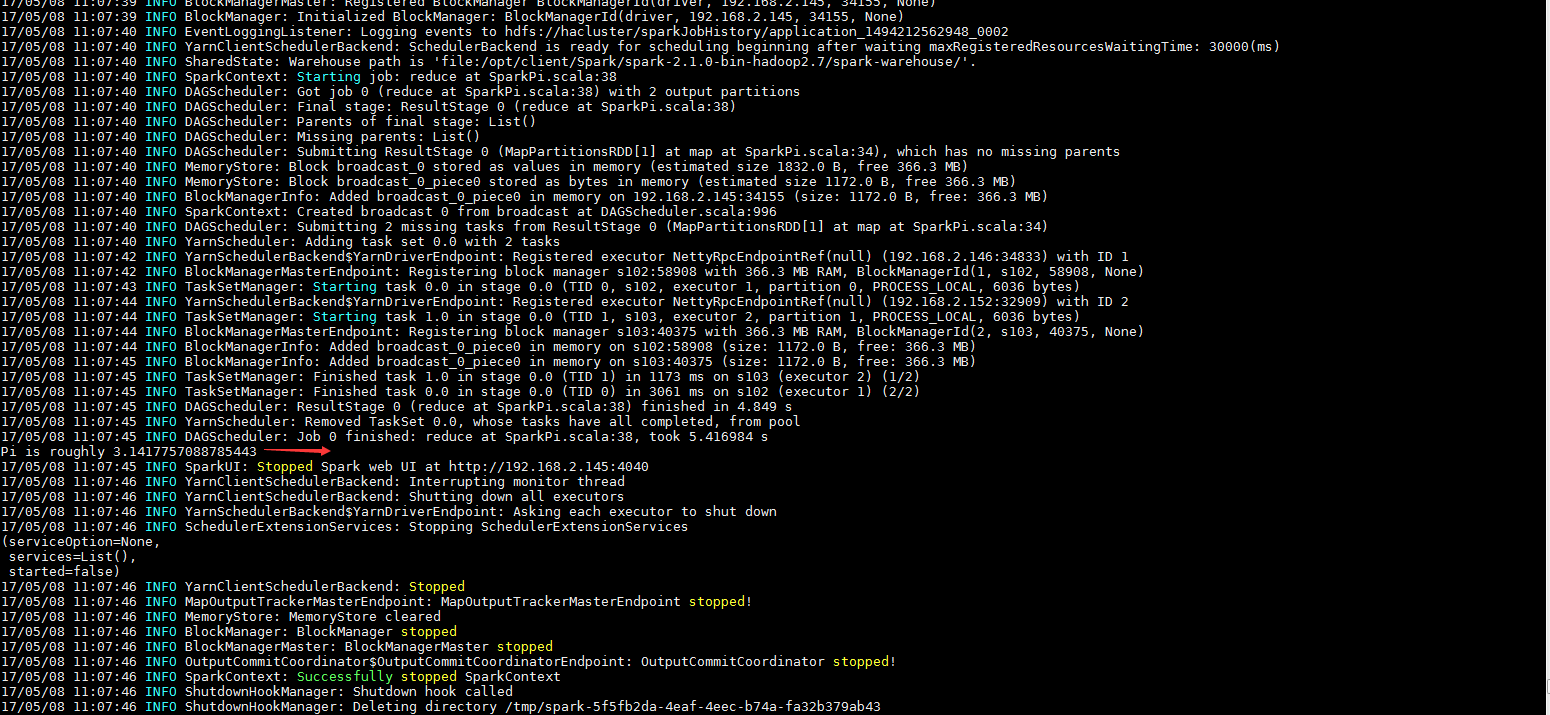
执行/usr/local/spark-2.2.0-bin-2.7.2/sbin/start-all.sh

验证是否启动成功

/usr/local/spark-2.2.0-bin-2.7.2/bin/spark-shell

/usr/local/spark-2.2.0-bin-2.7.2/bin/spark-submit --master yarn-client

--class org.apache.spark.examples.JavaSparkPi /usr/local/spark-2.2.0-bin-2.7.2/examples/jars/spark-examples\_2.20-2.2.0.jar



1. 启动spark ha

直接在s115执行/usr/local/spark-2.2.0-bin-2.7.2/sbin/start-master.sh

七．安装rocketMQ

1.规划服务器

规划三台虚拟机，一台安装NameServer，两台Master，没有slave。

NameServer服务器：172.18.18.102（hostname:s102）

两台Master服务器：172.18.18.100（hostname:s100）、172.18.18.101（hostname:s101）

2.开放rocketMQ端口（如果防火墙未关闭的话）

检查是否正确安装jdk，java –version

rocketMQ默认使用3个端口：9876、10911、10912 。

nameServer默认使用9876端口，master和slave默认使用10911端口、10912端口

开放端口的命令：/sbin/iptables -I INPUT -p tcp --dport 9876 -j ACCEPT

service iptables save  service iptables restart

3.配置/etc/hosts文件

在/etc/hosts文件末尾增加主机名和127.0.0.1的映射

127.0.0.1 localhost

4.安装RocketMQ（以下操作需要在三台机子上都执行一遍）

①、下载rocketMQ压缩包并上传到/opt/文件夹中

②、解压rocketMQ压缩包，如：tar –zxvf rocketmq-3.2.6.tar.gz，并将解压出来的文件夹更名为rocketmq。

③、配置环境变量：在/etc/profile文件最后添加： export ROCKETMQ\_HOME=/usr/local/rocketmq

export PATH=$PATH:$ROCKETMQ\_HOME/bin

export NAMESRV\_ADDR="172.18.18.114:9876"

添加完成后，执行source /etc/profile 命令使修改生效

④、修改文件权限：

cd rocketmq/bin/

chmod +x mqadmin mqbroker mqfiltersrv mqshutdown mqnamesrv

5.启动NameServer（在主节点上启动nameServer）

cd /opt/rocketmq/bin/

./mqnamesrv &

通过jps查看进程中是否启动NameServer

6.启动Broker（在两个Master上启动broker）

cd /opt/rocketmq/bin/

 ./mqbroker &

通过jps查看进程中是否启动

7.添加rocketmq web控制台的war包到tomcat底下

tomcat安装在NameServer所在的服务器，访问的地址为：[http://172.18.18.114:8089/rocketmq-console/cluster/list.do](http://172.18.18.102:8080/rocketmq-console/cluster/list.do)

8.rocketMQ具体使用:

http://jameswxx.iteye.com/blog/2091971

**进入/RocketMQ/bin目录下**，查看具体命令的使用：mqadmin help updateTopic

**查看当前集群列表**：mqadmin clusterList(单个nameServer情况下可以不加-n)

**多个NameServer情况下，**查看指令为：mqadmin clusterList –n 172.18.18.102:9876

**查看当前topic列表**：mqadmin topicList (单个nameServer情况下可以不加-n)

**多个NameServer情况下**，查看指令为：mqadmin topicList –n 172.18.18.102:9876

**创建Topic**：比如：updateTopic –b 172.18.18.100:10911 –n 172.18.18.102:9876 –t nihao 或者updateTopic –c DefaultCluster –n 172.18.18.102:9876 –t nihao

**用mqadmin topicList查看topic是否创建成功**

com.alibaba.rocketmq.tools.command.topic.UpdateTopicSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -b | 如果-c为空，则必填 | broker 地址，表示topic 建在该broker |
| -c | 如果-b为空，则必填 | cluster 名称，表示topic 建在该集群（集群可通过clusterList 查询） |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |
| -p | 否 | 指定新topic 的权限限制( W|R|WR ) |
| -r | 否 | 可读队列数（默认为8） |
| -w | 否 | 可写队列数（默认为8） |
| -t | 是 | opic 名称（名称只能使用字符 ^[a-zA-Z0-9\_-]+$ ） |

**删除topic**：mqadmin deleteTopic –c DefaultCluster –n 172.18.18.102:9876 –t woshishabi

类路径

com.alibaba.rocketmq.tools.command.topic.DeleteTopicSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -c | 是 | cluster 名称，表示删除某集群下的某个topic （集群可通过clusterList 查询） |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;… |
| -t | 是 | topic 名称（名称只能使用字符 ^[a-zA-Z0-9\_-]+$ ） |

**查看topic的路由信息**：mqadmin topicRoute –t woshishabi –n 172.18.18.102:9876

类路径

com.alibaba.rocketmq.tools.command.topic.TopicRouteSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -t | 是 | topic 名称 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

**查看topic的统计信息**：mqadmin topicStatus -t woshishabi -n 172.18.18.102:9876

类路径

com.alibaba.rocketmq.tools.command.topic.TopicStatsSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -t | 是 | topic 名称 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

**查看Broker统计信息**：mqadmin brokerStatus –b 172.18.18.100:10911 –n 172.18.18.102:9876

类路径

com.alibaba.rocketmq.tools.command.broker.BrokerStatsSubCommanD

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -b | 是 | broker 地址 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

**创建（修订）订阅组：**mqadmin updateSubGroup –c DefaultCluster –g woshiGroup –n 172.18.18.102:9876

类路径

com.alibaba.rocketmq.tools.command.consumer.UpdateSubGroupSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -b | 如果 –c 为空，则必填 | broker 地址，表示订阅组建在该broker |
| -c | 如果 –b 为空，则必填 | cluster名称，表示topic 建在该集群（集群可通过clusterList查询） |
| -d | 否 | 是否容许广播方式消费 |
| -g | 是 | 订阅组名 |
| -i | 否 | 从哪个broker 开始消费 |
| -m | 否 | 是否容许从队列的最小位置开始消费，默认会设置为false |
| -q | 否 | 消费失败的消息放到一个重试队列，每个订阅组配置几个重试队列 |
| -r | 否 | 重试消费最大次数，超过则投递到死信队列，不再投递，并报警 |
| -s | 否 | 消费功能是否开启 |
| -w | 否 | 发现消息堆积后，将Consumer 的消费请求重定向到另外一台Slave 机器 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

**删除订阅组**：mqadmin deleteSubGroup –c DefaultCluster –g woshiGroup –n 172.18.18.102:9876

类路径

com.alibaba.rocketmq.tools.command.consumer.DeleteSubscriptionGroupCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -b | 如果–c 为空，则必填 | broker 地址，表示订阅组建在该broker |
| -c | 如果–b 为空，则必填 | cluster 名称，表示topic建在该集群（集群可通过clusterList查询） |
| -g | 是 | 订阅组名 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

**更新broker配置文件**：

指令

updateBrokerConfig

类路径

com.alibaba.rocketmq.tools.command.broker.UpdateBrokerConfigSubCommand

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必填 | 说明 |
| -b | 如果–c为空，则必填 | broker 地址，表示订阅组建在该broker |
| -c | 如果–b 为空，则必填 | cluster名称，表示topic 建在该集群（集群可通过clusterList查询） |
| -k | 是 | key 值 |
| -v | 否 | value 值 |
| -h | 否 | 打印帮助 |
| -n | 是 | nameserve 服务地址列表，格式ip:port;ip:port;... |

八．安装ElasticSearch搜索引擎

1.先安装java

2、检查系统中是否已经安装了java：

Java –version

3、安装JDK

解压命令：tar –xvf jdk-8u131-linux-x64.tar.gz

在/etc/profile中配置环境变量：export JAVA\_HOME=/usr/local/jdk

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

配置完成后：source /etc/profile

4.ElasticSearch的安装

5、将es压缩文件移到/usr/local中

移动命令：mv elasticsearch-2.1.1.tar.gz /opt

6、解压并重命名

命令：tar –xvf elasticsearch-2.1.1.tar.gz mv elasticsearch-2.1.1.tar.gz elasticsearch

7、因为用root用户运行es会报错，所以为了系统安全，需要单独创建一个用户使用elasticsearch

命令：groupadd elsearch useradd elsearch –g elsearch –p elasticsearch

8、更改elasticsearch文件夹及内部文件的所属用户及组为elsearch:elsearch

命令：chown -R elsearch:elsearch elasticsearch

1. 修改 vim /opt/elastic/config/elasticsearch.yml

在s114上添加以下内容

cluster.name: es-cluster

node.name: s114

network.host: 172.18.18.114

discovery.zen.ping.unicast.hosts: ["s114", "s115", "s116"]

在s115上添加以下内容

cluster.name: es-cluster

node.name: s115

network.host: 172.18.18.115

discovery.zen.ping.unicast.hosts: ["s114", "s115", "s116"]

在s116上添加以下内容

cluster.name: es-cluster

node.name: s116

network.host: 172.18.18.116

discovery.zen.ping.unicast.hosts: ["s114", "s115", "s116"]

10.如果三台节点下/opt/elastic/data目录有内容，把data目录清空

11、切换到elsearch用户再启动

命令：su elsearch cd elasticsearch/bin ./elasticsearch

10、因为elasticsearch启动后会一直存在在界面上，所以可以让它在后台运行

命令：./elasticsearch -d

九．安装hive

1. 安装mysql

yum -y install mysql mysql-server mysql-devel（安装系统自带）

service mysqld start  //开启服务

chkconfig mysqld on //开机自启

cat /root/.mysql\_secret(查看隐藏文件，看mysql密码)

mysql -uroot -pgN8h7tIJiQFfo5Mn //登录mysql

mysql>set password=password('123456');//修改密码

Mysql>grant all on \*（所有数据库）.\*（所有表） to root@'s114'(客户端) identified by '123456' ;  //授权

Mysql>grant all on \*（所有数据库）.\*（所有表） to root@'s115'(客户端) identified by '123456' ;  //授权

Mysql>grant all on \*（所有数据库）.\*（所有表） to root@'s116'(客户端) identified by '123456' ;  //授权

mysql> flush privileges ;

Mysql>quit;//退出

1. 下载hive的安装包，上传至/usr/local,cd /usr/local，解压安装，tar -zxvf

apache-hive-bin.tar.gz

1. 修改配置文件

cd /usr/local/apache-hive-bin/conf

mv hive-env.sh.template hive-env.sh

vim hive-env.sh，添加以下内容

export JAVA\_HOME=/opt/tool/jdk

export HIVE\_HOME=/usr/local/apache-hive-1.1.0-bin

export HADOOP\_HOME=/usr/local/hadoop-2.7.2

cp -a hive-default.xml.template hive-site.xml

vim hive-site.xml

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://s114:3306/hive?createDatabaseIfNotExist=true&amp;useUnicode=true&amp;characterEncoding=latin1</value>

<description>JDBC connect string for a JDBC metastore</description>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

<description>Driver class name for a JDBC metastore</description>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

<description>username to use against metastore database</description>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

<description>password to use against metastore database</description>

</property>

<property>

<name>hive.server2.thrift.port</name>

<value>10000</value>

</property>

<property>

<name>hive.server2.thrift.bind.host</name>

<value>s115</value>

</property>

3.拷贝驱动包到$HIVE\_HOME/lib

cp mysql-connector-java-5.1.27.jar /usr/local/hive/lib

1. 通过Hivemetastore，HiveServer使用hive

nohup /usr/local/apache-hive-1.1.0-bin/bin/hive --service metastore >/dev/null 2>&1 &

nohup /usr/local/apache-hive-1.1.0-bin/bin/hive --service hiveserver2 >/dev/null 2>&1 &

bin/beeline

beeline>!connect jdbc:hive2://s114:10000 （连接服务端命令）

        输入用户名账号和密码

十．安装kafka

1.下载scala-2.11.8.tgz，上传至目录/usr/local，cd /usr/local

解压tar -zxvf scala-2.11.8.tgz

2.chmod -R 755 /usr/local/scala-2.11.8，分发至其他节点

scp -r /usr/local/scala-2.11.8 s115:/usr/local

scp -r /usr/local/scala-2.11.8 s116:/usr/local

3.每台节点都配置系统变量

vim /etc/profile //在文件最后加上以下内容

export SCALA\_HOME=/usr/local/scala-2.11.8

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

source /etc/profile //更新

4.验证scala

scala -version //出现scala的信息

5.确保zookeeper安装并启动成功

6.下载kafka\_2.11-0.10.2.0.tgz，上传至目录/usr/local，cd /usr/local

解压tar -zxvf kafka\_2.11-0.10.2.0.tgz

1. chmod -R 755 kafka\_2.11-0.10.2.0 //修改权限

8.修改配置文件/usr/local/kafka\_2.11-0.10.2.0/conf目录下server.properties文件

其中broker.id不能重复，分发后需做修改

指定log.dirs后，需创建相应路径

//s114上对应broker.id为0，s115上对应broker.id为1，s116上对应broker.id为2

//s114上对应listeners 为PLAINTEXT://s114:9092，s115上对应listeners 为PLAINTEXT://s115:9092，s116上对应listeners 为PLAINTEXT://s116:9092

broker.id=0

listeners = PLAINTEXT://s114:9092

#advertised.listeners=PLAINTEXT://your.host.name:9092

num.network.threads=3

num.io.threads=8

socket.send.buffer.bytes=102400

socket.receive.buffer.bytes=102400

socket.request.max.bytes=104857600

log.dirs=/usr/local/kafka\_2.11-0.10.2.0/tmp

num.partitions=1

num.recovery.threads.per.data.dir=1

#log.flush.interval.ms=1000

log.retention.hours=168

#log.retention.bytes=1073741824

log.segment.bytes=1073741824

log.retention.check.interval.ms=300000

zookeeper.connect=s114:2181,s115:2181,s116:2181

zookeeper.connection.timeout.ms=6000

1. 在s114上分发安装包到s115和s116

scp -r /usr/local/kafka\_2.11-0.10.2.0 s115:/usr/local

修改conf目录下的server.properties文件的broker.id=1

修改conf目录下的server.properties文件的listeners 为PLAINTEXT://s115:9092

scp -r /usr/local/kafka\_2.11-0.10.2.0 s116:/usr/local

修改conf目录下的server.properties文件的broker.id=2

修改conf目录下的server.properties文件的listeners 为PLAINTEXT://s116:9092

10.分发后，在各个节点逐一启动kafka

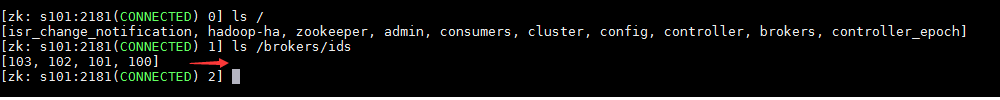
$> kafka-server-start.sh /opt/client/Kafka/kafka\_2.11-0.10.2.0/config/server.properties

必须指定配置文件

11..通过zkCli.sh查看broker注册情况

$>zkCli.sh -server s101:2181

$>ls /brokers/ids



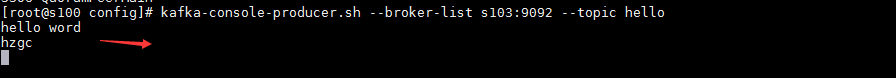
12.测试是否可以使用

【创建主题topic】

$>kafka-topics.sh --create --zookeeper s101:2181 --replication-factor 1 --partitions 1 --topic hello

【控制台输入（producer）】

$>kafka-console-producer.sh --broker-list s103:9092 --topic hello



【控制台接收（consumer）】

