大数据环境安装

2016年5月30日 星期一

11:40

|  |  |  |  |
| --- | --- | --- | --- |
| IP | 主机名 | 用户名 | 功能 |
| 133.37.31.80 | scdbdatanode27 | root  hadoop | ResourceManager，NameNode，flume，zk，kafka，hive，Presto |
| 133.37.31.81 | scdbdatanode28 | root  hadoop | NodeManager，DataNode，JournalNode，flume，zk，kafka ，hive，Presto |
| 133.37.31.82 | scdbdatanode29 | root  hadoop | NodeManager，DataNode，JournalNode，flume，zk，kafka，hive，Presto |
| 133.37.31.83 | scdbdatanode30 | root  hadoop | NodeManager，DataNode，JournalNode，flume，kafka，hive，Presto |
| 133.37.31.84 | scdbdatanode31 | root  hadoop | NodeManager，DataNode，JournalNode，flume，kafka，hive，Presto |
| 133.37.31.85 | scdbdatanode32 | root  hadoop | NodeManager，DataNode，JournalNode，flume，kafka，hive，Presto |

公共应用目录，比如jdk

/usr/local/jdk1.6.0\_39

应用安装目录

/app

hadoop相关存储目录

/hadoop/data1/zkdata

/hadoop/data1/yarndata

/hadoop/data1/tmp

/hadoop/data1/name

/hadoop/data1/pids

/hadoop/data1/journal

/hadoop/data1/presto

/hadoop/data1/zklogs

/hadoop/data1/yarnlogs

/hadoop/data1/logs

/hadoop/data1/kafka-logs

/hadoop/data2/data

/hadoop/data3/data

/hadoop/data4/data

/hadoop/data5/data

/hadoop/data6/data

/hadoop/data7/data

/hadoop/data8/data

/hadoop/data9/data

/hadoop/data10/data

/hadoop/data11/data

/hadoop/data12/data

# 准备工作

## 1、修改主机名

#vi /etc/sysconfig/network

scdbdatanode27~ scdbdatanode32

计算机生成了可选文字:
Croot@localhost 
Croot@localhost 
NETWORKING=yes 
HOSTNAME=bg1 
Croot@localhost 
vi /etc/sysconfig/network 
cat /etc/sysconfig/network 

## 2、创建用户

创建用户、修改密码（root用户，所有机器上都操作）

#groupadd hadoop

#useradd hadoop -g hadoop

#passwd hadoop

计算机生成了可选文字:
Croot@localhost groupadd hadoop 
Croot@localhost useradd hadoop -g hadoop 
Croot@localhost passwd hadoop 
hadoop 
*fifi hadoop 
passwd: 
Croot@localhost 

查看所有用户组：

#cat /etc/group

查看用户组下的用户：

#groups groupname

## 3、修改host

(之前需要修改主机名 #vi /etc/sysconfig/network)

修改 /etc/hosts文件内容（root用户，所有机器上都操作）

所有机器的/etc/hosts都修改成下面内容

*127.0.0.1 localhost*

*133.37.31.80 scdbdatanode27*

*133.37.31.81 scdbdatanode28*

*133.37.31.82 scdbdatanode29*

*133.37.31.83 scdbdatanode30*

*133.37.31.84 scdbdatanode31*

*133.37.31.85 scdbdatanode32* 计算机生成了可选文字:
Croot@localhost cat /etc/hosts 
localhost localhost. localdomain localhost4 localhost4. localdomain4 
127.0.0.1 
localhost localhost. localdomain localhost6 localhost6. localdomain6 
192.168.33.11 bgl 
192.168.33.12 bg2 
192.168.33.13 bg3 
192.168.33.14 bg4 
Croot@localhost 

## 4、关闭防火墙

$service iptables stop （临时关闭）

$chkconfig iptables off （重启后生效）

计算机生成了可选文字:
Croot@localhost service iptables stop 
Croot@localhost chkconfig iptables off 
Croot@localhost 

## 5、配置ssh登录

（为了便于维护，建议root、hadoop用户均处理）

5.1、在主节点上执行以下命令生成密钥对

#ssh-keygen -t rsa

5.2、将公钥添加到认证文件中：

#cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

5.3、设置authorized\_keys的访问权限：

#chmod 600 ~/.ssh/authorized\_keys

5.4、scp文件到所有datenode节点：

#scp ~/.ssh/authorized\_keys root@scdbdatanode28:~/.ssh/

计算机生成了可选文字:
Croot@localhost ssh-keygen -t rsa 
Generating public/private rsa key pair. 
Enter file in which to save the key C/ root/ .ssh/id_rsa): 
Created directory ' / root/ .ssh' . 
Enter passphrase (empty for no passphrase): 
Enter same passphrase again: 
Your identification has been saved in / root/ .ssh/id_rsa. 
Your public key has been saved in / root/ .ssh/id_rsa .pub. 
The key fingerprint is: 
root@localhost.localdomain 
The key's randomart image is: 
+ — — RSA — 
00. 
.00. B.. 
S.O.E 
Croot@localhost 
Croot@localhost 
Croot@localhost 
The authenticity 
cat N/.ssh/id_rsa.pub N/.ssh/authorized_keys 
chmod 600 N/.ssh/authorized_keys 
ssh root@bgl 
of host 'bgl (192.168.33.11)' can't be established. 
RSA key fingerprint is 5c:97:4b:96:a2:41:a8:44:cc:70:b1:5e:8d:a7:a5:3b. 
Are you sure you want to continue connecting (yes/no)? yes 
Warning: Permanently added 'bg1,192.168.33.11' (RSA) to the list of known hosts. 
Last login: Mon May 30 2016 from 192.168.33.1 
Croot@localhost ssh root@bgl 
Last login: Mon May 30 04:52:49 2016 from bgl 

最好互相打通，依次在每一个机器上4.1、4.2、4.3、4.4，每次在执行4.4的时候，scp的目标主机取一台新主机，直到完成最后一台后，再在最后一台主机上scp到其他所有主机，可以保证整个集群两两互通。

## 6、配置ntp

安装ntp

#yum install ntp

配置开机启动

#chkconfig ntpd on

检查是否成功

#chkconfig --list ntpd

其中2-5为on状态就代表成功。

计算机生成了可选文字:
Croot@bg4 
Croot@bg4 
ntpd 
chkconfig 
chkconfig 
ntpd on 
-- list ntpd 
5•.EX 

修改时区

#cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

查看时区

计算机生成了可选文字:
Croot@bgl hadoop]# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime 
cp: "/etc/localtime"? 
Croot@bgl hadoop]# date -R 
Tue, 31 May 2016 22:49: 1 +0800 

测试环境能访问外网同步，暂时不配置局域网同步

## 7、安装Oracle的JDK

卸载openJDK

#rpm -qa | grep java

#yum -y remove java-1.5.0-gcj-1.5.0.0-29.1.el6.x86\_64

去Oracle的官网下载jdk的rpm安装包，并使用rpm -ivh 包名安装。如果是tar.gz文件采用 tar -xvzf /vagrant/jdk1.6.0\_39.tar.gz -C /usr/local/ 命令

way1:

#vi /etc/profile

export JAVA\_HOME=/usr/local/jdk1.6.0\_39

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

way2:

#echo "export JAVA\_HOME=/usr/local/jdk1.6.0\_39" >> /etc/profile

#echo "export PATH=\$JAVA\_HOME/bin:\$PATH" >> /etc/profile

如果提示 #java -versin 没有权限，则改变执行权限

#chmod 551 /usr/local/jdk1.6.0\_39/bin/java

#chmod 551 /usr/local/jdk1.6.0\_39/bin/jps

加载环境变量

#source /etc/profile

执行命令验证

#java -version

## 8、准备目录权限

#chown -R hadoop:hadoop /hadoop/

#chown -R hadoop:hadoop /app/

以下操作如果没有特殊说明，均切换成hadoop用户操作

# 安装ZK3.4.5

## 1、解压文件

#tar -zxvf /zookeeper-3.4.5.tar.gz -C /app/

## 2、配置zoo.cfg

#cp zoo\_sample.cfg zoo.cfg

计算机生成了可选文字:
Chadoop@bgl conf]$ pwd 
/app/zookeeper-3.4.5/ conf 
Chadoop@bgl conf]$ cp zoo_sample. cfg zoo. cfg 
Chadoop@bgl conf]$ 

#vi zoo.cfg

*tickTime=2000*

*initLimit=10*

*syncLimit=5*

*dataDir=/hadoop/data1/zkdata*

*clientPort=2181*

*server.1= scdbdatanode27:2888:3888*

*server.2= scdbdatanode28:2888:3888*

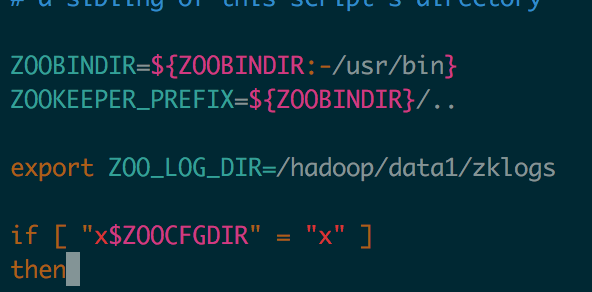
*server.3= scdbdatanode29:2888:3888*

## 3、设置ZK的日志路径

#vi /app/zookeeper-3.4.5/bin/zkEnv.sh

设置值：

*export ZOO\_LOG\_DIR=/hadoop/data1/zklogs*



## 4、拷贝配置文件到其他节点

#scp /app/zookeeper-3.4.5/conf/zoo.cfg scdbdatanode28:/app/zookeeper-3.4.5/conf/

#scp /app/zookeeper-3.4.5/conf/zoo.cfg scdbdatanode29:/app/zookeeper-3.4.5/conf/

#scp /app/zookeeper-3.4.5/bin/zkEnv.sh scdbdatanode28:/app/zookeeper-3.4.5/bin/

#scp /app/zookeeper-3.4.5/bin/zkEnv.sh scdbdatanode29:/app/zookeeper-3.4.5/bin/

也可以在第一个节点完成配置后整个目录进行scp –r

## 5、创建myid

scdbdatanode27#echo 1 > /hadoop/data1/zkdata/myid

scdbdatanode28#echo 2 > /hadoop/data1/zkdata/myid

scdbdatanode29#echo 3 > /hadoop/data1/zkdata/myid

## 6、追加环境变量

（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export ZOOKEEPER\_HOME=/app/zookeeper-3.4.5" >> /etc/profile

#echo "export PATH=\$ZOOKEEPER\_HOME/bin:\$PATH" >> /etc/profile

加载环境变量（hadoop用户）

#source /etc/profile

## 7、各节点启动

#zkServer.sh start | stop

检查状态

计算机生成了可选文字:
Chadoop@bgl bin] 
$ jps 
2442 QuorumPeerMain 
2467 JPS 
Chadoop@bgl bin]$ zkServer. sh status 
JMX enabled by default 
Using config: /app/zookeeper-3.4.5/bin/. ./conf/zoo. cfg 
Mode: follower 

计算机生成了可选文字:
Chadoop@bg2 jps 
2403 JPS 
2363 QuorumPeerMain 
Chadoop@bg2 zkServer. sh status 
JMX enabled by default 
Using config: /app/zookeeper-3.4.5/bin/. ./conf/zoo. cfg 
Mode: follower 

计算机生成了可选文字:
Chadoop@bg3 jps 
2353 QuorumPeerMain 
2400 JPS 
Chadoop@bg3 zkServer. sh status 
JMX enabled by default 
Using config: /app/zookeeper-3.4.5/bin/.. / conf/ zoo. cfg 
Mode: leader 

## 8、编辑zkClient工程验证

计算机生成了可选文字:
App [Java Application] [Library/Java/JavaVirtualMachines/jdk1.8.0_91 .jdk/Contents/Home/bin/java (2016#6E 1 a 
log4j : WARN 
log4j : WARN 
log4j : WARN 
log4j : WARN 
No appenders could be 
found for logger (org.I@Itec.zkclient.ZkConnection) . 
No appenders could be 
found for logger (org.IØItec.zkclient.ZkEventThread). 
Please initialize the 
log4j system properly. 
Please initialize the 
log4j system properly. 
uuid string, 'uuidStr 
'--->44cac633-b8a1-4d51-8Ø53-9Ø8edc7a7516 

计算机生成了可选文字:
App 
Java Application] /LlbrarY/Java/JavaVlrtualMacmnes/Jdk 
d 0 91 .Jdk/contents/Home/bln/java 
log4j :WARN 
No appenders could be found 
for logger (org.IØItec.zkclient.ZkEventThread). 
log4j : WARN 
No appenders could be found 
for logger (org.IØItec.zkclient.ZkConnection). 
log4j : WARN 
Please initialize the log4j 
system properly. 
log4j : WARN 
Please initialize the log4j 
system properly. 
the node 'dataPath'===>/root1, data has changed. it's data is 44cac633-b8a1-4d51-8Ø53-9Ø8edc7a7516 
the node 'dataPath'===>/root1, data has changed. it's data is 41f9daed-7ebb-4da6-975b-e6ed4b1142Ø8 

PS：zkClient zk客户端工具，后续学习

参考： <http://www.cnblogs.com/yql1986/p/4116483.html>

# 安装hadoop2.2.0

配置高可用环境详细解读 <http://www.linuxidc.com/Linux/2014-09/106289.htm>

## 1、下载并解压

[hadoop@scdbdatanode27 ~]$ tar -xzvf /app/hadoop-2.2.0.tar.gz -C /app/

## 2、追加环境变量

（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export HADOOP\_HOME=/app/hadoop-2.2.0" >> /etc/profile

#echo "export PATH=\$HADOOP\_HOME/bin:\$HADOOP\_HOME/sbin:\$PATH" >> /etc/profile

加载环境变量（hadoop用户）

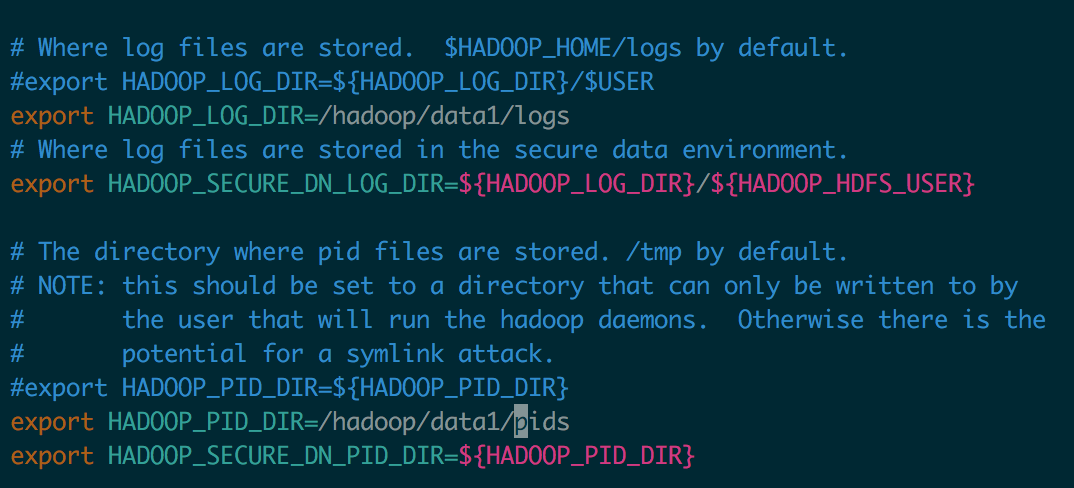
#source /etc/profile

## 3、修改hadoop、yarn日志目录

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/hadoop-env.sh

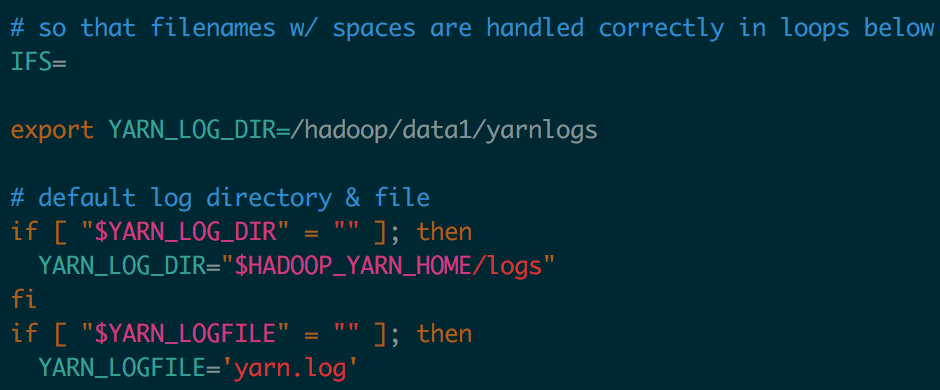
*export HADOOP\_LOG\_DIR=/hadoop/data1/logs*

*export HADOOP\_PID\_DIR=/hadoop/data1/pids*



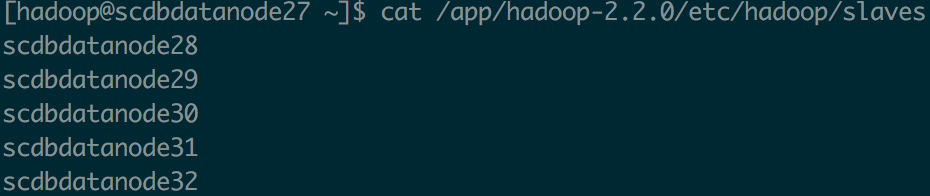
[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/yarn-env.sh

*export YARN\_LOG\_DIR=/hadoop/data1/yarnlogs*

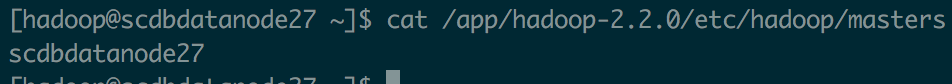


## 4、配置masters和slaves

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/slaves



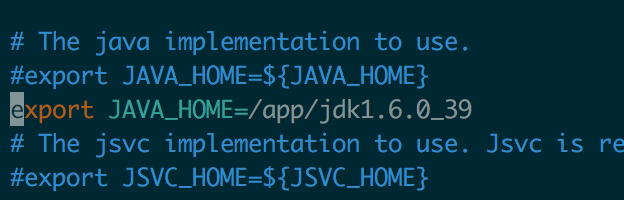
[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/masters



## 5、修改hadoop-env.sh中的JAVA\_HOME

避免后续出现不可预料的异常，必须修改此配置。

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/hadoop-env.sh



## 6、修改 /app/hadoop-2.2.0/etc/hadoop/core-site.xml

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/core-site.xml

*<configuration>*

*<property>*

*<name>fs.defaultFS</name>*

*<value>hdfs://scdbdatanode27:9010</value>*

*<!--*

*以上配置的scdbdatanode27是固定host，只适用于手动切换主备NameNode的场景*

*如果需要通过ZooKeeper来自动切换，则需要配置逻辑名称*

*此host名称需要同hdfs-site.xml中 dfs.nameservices key 的value值保持一致*

*port需要和在hdfs-site定义的RPC端口一致*

*-->*

*</property>*

*<property>*

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

*<value>*/hadoop/data1/tmp*</value>*

*</property>*

*</configuration>*



## 7、修改/app/hadoop-2.2.0/etc/hadoop/hdfs-site.xml

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/hdfs-site.xml

*<configuration>*

*<property>*

*<name>dfs.nameservices</name>*

*<value>nameservice1</value>*

*<description>*

*指定命名空间名称，可随意起名*

*dfs.nameservices key 的value值需要同core-site.xml中fs.defaultFS中host保持一致*

*Comma-separated list of nameservices.*

*</description>*

*</property>*

*<property>*

*<name>dfs.ha.namenodes.nameservice1</name>*

*<value>nn1,nn2</value>*

*<description>*

*在命名空间下指定NameNode逻辑名*

*The prefix for a given nameservice, contains a comma-separated*

*list of namenodes for a given nameservice (eg EXAMPLENAMESERVICE).*

*</description>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.nameservice1.nn1</name>*

*<value>scdbdatanode27:9010</value>*

*<description>*

*为“命名空间名.NameNode逻辑名”配置rpc地址*

*需要同core-site.xml中fs.defaultFS的Port保持一致*

*RPC address for nomenode1 of hadoop-test*

*</description>*

*</property>*

*<property>*

*<name>dfs.namenode.rpc-address.nameservice1.nn2</name>*

*<value>scdbdatanode28:9010</value>*

*<description>*

*为“命名空间名.NameNode逻辑名”配置rpc地址*

*需要同core-site.xml中fs.defaultFS的Port保持一致*

*RPC address for nomenode2 of hadoop-test*

*</description>*

*</property>*

*<property>*

*<name>dfs.namenode.http-address.nameservice1.nn1</name>*

*<value>scdbdatanode27:50070</value>*

*<description>*

*为“命名空间名.NameNode逻辑名”配置http地址*

*The address and the base port where the dfs namenode1 web ui will listen on.*

*</description>*

*</property>*

*<property>*

*<name>dfs.namenode.http-address.nameservice1.nn2</name>*

*<value>scdbdatanode28:50070</value>*

*<description>*

*为“命名空间名.NameNode逻辑名”配置http地址*

*The address and the base port where the dfs namenode2 web ui will listen on.*

*</description>*

*</property>*

*<property>*

*<name>dfs.namenode.name.dir</name>*

*<value>/hadoop/data1/name</value>*

*<description>*

*配置NameNode元数据存放的路径；*

*如果机器上有多块硬盘的话，推荐配置多个路径，用逗号分隔。*

*Determines where on the local filesystem the DFS name node*

*should store the name table(fsimage). If this is a comma-delimited list*

*of directories then the name table is replicated in all of the*

*directories, for redundancy.*

*</description>*

*</property>*

*<property>*

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

*<value>qjournal://scdbdatanode28:8485;scdbdatanode29:8485;scdbdatanode30:8485;scdbdatanode31:8485;scdbdatanode32:8485/nameservice1</value>*

*<description>*

*配置JournalNode，包含三部分：*

*（1）qjournal是协议，无需修改；*

*（2）然后就是三台部署JournalNode的主机host/ip：端口，三台机器之间用分号分隔；*

*（3）最后的hadoop-journal是journalnode的命名空间，可以随意取名。*

*A directory on shared storage between the multiple namenodes*

*in an HA cluster. This directory will be written by the active and read*

*by the standby in order to keep the namespaces synchronized. This directory*

*does not need to be listed in dfs.namenode.edits.dir above. It should be*

*left empty in a non-HA cluster.*

*</description>*

*</property>*

*<property>*

*<name>dfs.datanode.data.dir</name>*

*<value>/hadoop/data2/data,/hadoop/data3/data,/hadoop/data4/data,/hadoop/data5/data,/hadoop/data6/data,/hadoop/data7/data,/hadoop/data8/data,/hadoop/data9/data,/hadoop/data10/data,/hadoop/data11/data,/hadoop/data12/data</value>*

*<description>*

*配置DataNode数据存放的路径；*

*如果机器上有多块硬盘的话，推荐配置多个路径，用逗号分隔。*

*Determines where on the local filesystem an DFS data node*

*should store its blocks. If this is a comma-delimited*

*list of directories, then data will be stored in all named*

*directories, typically on different devices.*

*Directories that do not exist are ignored.*

*</description>*

*</property>*

*<property>*

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

*<value>false</value>*

*<description>*

*是否自动切换。由于没有配置ZooKeeper，所以不能实现自动切换，所以这里配置的是false。*

*Whether automatic failover is enabled. See the HDFS High*

*Availability documentation for details on automatic HA configuration.*

*</description>*

*</property>*

*<property>*

*<name>dfs.journalnode.edits.dir</name>*

*<value>/hadoop/data1/journal/</value>*

*<description>*

*journalnode的本地数据存放目录，指定一个路径就够。*

*</description>*

*</property>*

*</configuration>*

## 8、修改/app/hadoop-2.2.0/etc/hadoop/mapred-site.xml

[hadoop@scdbdatanode27 ~]$ cp /app/hadoop-2.2.0/etc/hadoop/mapred-site.xml.template /app/hadoop-2.2.0/etc/hadoop/mapred-site.xml

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/mapred-site.xml

*<configuration>*

*<property>*

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

*<value>yarn</value>*

*</property>*

*<!--*

*jobhistory properties*

*jobhistory server，可以通过它查看已经运行完的应用程序的信息。*

*-->*

*<property>*

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

*<value>scdbdatanode28:10020</value>*

*<description>MapReduce JobHistory Server IPC host:port</description>*

*</property>*

*<property>*

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

*<value>scdbdatanode28:19888</value>*

*<description>MapReduce JobHistory Server Web UI host:port</description>*

*</property>*

*</configuration>*

## 9、修改/app/hadoop-2.2.0/etc/hadoop/yarn-site.xml

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/yarn-site.xml

*<configuration>*

*<property>*

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

*<value>scdbdatanode27</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.address</name>*

*<value>${yarn.resourcemanager.hostname}:8032</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.scheduler.address</name>*

*<value>${yarn.resourcemanager.hostname}:8030</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.webapp.address</name>*

*<value>${yarn.resourcemanager.hostname}:8088</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.webapp.https.address</name>*

*<value>${yarn.resourcemanager.hostname}:8090</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.resource-tracker.address</name>*

*<value>${yarn.resourcemanager.hostname}:8031</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.admin.address</name>*

*<value>${yarn.resourcemanager.hostname}:8033</value>*

*</property>*

*<property>*

*<name>yarn.resourcemanager.scheduler.class</name>*

*<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>*

*<description>*

*指定fairscheduler调度器*

*The class to use as the resource scheduler.*

*</description>*

*</property>*

*<property>*

*<name>yarn.scheduler.fair.allocation.file</name>*

*<value>${yarn.home.dir}/etc/hadoop/fairscheduler.xml</value>*

*<description>*

*指定fairscheduler调度器配置文件路径*

*fair-scheduler conf location*

*</description>*

*</property>*

*<property>*

*<name>yarn.nodemanager.local-dirs</name>*

*<value>/hadoop/data1/yarndata</value>*

*<description>*

*指定nodemanager的本地工作目录，推荐配置多个路径，用逗号分隔*

*List of directories to store localized files in. An*

*application's localized file directory will be found in:*

*${yarn.nodemanager.local-dirs}/usercache/${user}/appcache/application\_${appid}.*

*Individual containers' work directories, called container\_${contid}, will*

*be subdirectories of this.*

*</description>*

*</property>*

*<property>*

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

*<value>true</value>*

*</property>*

*<property>*

*<name>yarn.nodemanager.remote-app-log-dir</name>*

*<value>/hadoop/data1/yarnlogs</value>*

*</property>*

*<property>*

*<name>yarn.nodemanager.resource.memory-mb</name>*

*<value>2048</value>*

*<description>*

*每个nodemanager上可以用的内存大小*

*Amount of physical memory, in MB, that can be allocated for containers.*

*注意：我的NM虚拟机是1G内存，1核CPU，当该值配置小于1024时，NM是无法启动的！会报错：*

*NodeManager from slavenode2 doesn't satisfy minimum allocations, Sending SHUTDOWN signal to the NodeManager.*

*</description>*

*</property>*

*<property>*

*<name>yarn.nodemanager.resource.cpu-vcores</name>*

*<value>4</value>*

*<description>*

*每个nodemanager上可用的CPU核数*

*Number of CPU cores that can be allocated for containers.*

*</description>*

*</property>*

*<property>*

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

*<value>mapreduce\_shuffle</value>*

*<description>*

*the valid service name should only contain a-zA-Z0-9\_ and can not start with numbers*

*</description>*

*</property>*

*</configuration>*

## 10、创建/app/hadoop-2.2.0/etc/hadoop/fairscheduler.xml

[hadoop@scdbdatanode27 ~]$ vi /app/hadoop-2.2.0/etc/hadoop/fairscheduler.xml

*<?xml version="1.0"?>*

*<allocations>*

*<queue name="infrastructure">*

*<minResources>1024 mb, 8 vcores</minResources>*

*<maxResources>2048 mb, 16 vcores</maxResources>*

*<maxRunningApps>20</maxRunningApps>*

*<minSharePreemptionTimeout>30</minSharePreemptionTimeout>*

*<weight>1.0</weight>*

*<aclSubmitApps>root,hadoop,yarn</aclSubmitApps>*

*</queue>*

*<queue name="tool">*

*<minResources>1024 mb, 8 vcores</minResources>*

*<maxResources>2048 mb, 16 vcores</maxResources>*

*</queue>*

*</allocations>*

## 11、启动集群

以下操作均试用hadoop用户进行

11.1、在各个JournalNode节点上，输入以下命令启动journalnode服务：

（scdbdatanode28~ scdbdatanode32）

#hadoop-daemon.sh start journalnode

11.2、在[nn1]（scdbdatanode27）上，对其进行格式化，并启动：

#hdfs namenode –format(仅初次启动的时候执行一次)

#hadoop-daemon.sh start namenode

11.3、在[nn2]（scdbdatanode28）上，同步nn1的元数据信息：

#hdfs namenode -bootstrapStandby

11.4、启动[nn2]（scdbdatanode28）：

#hadoop-daemon.sh start namenode

经过以上四步操作，nn1和nn2均处理standby状态

11.5、将[nn1]（scdbdatanode27）切换为Active

#hdfs haadmin -transitionToActive nn1

11.6、在[nn1]（scdbdatanode27）上，启动所有datanode

#hadoop-daemons.sh start datanode

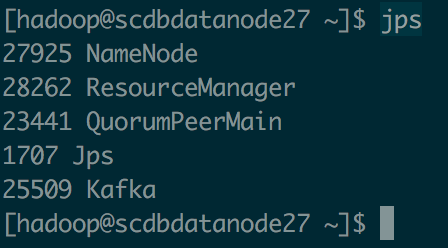
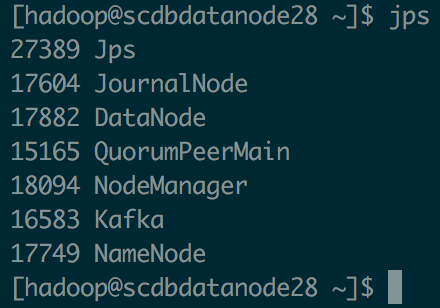
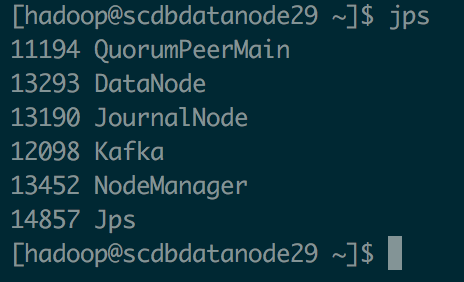
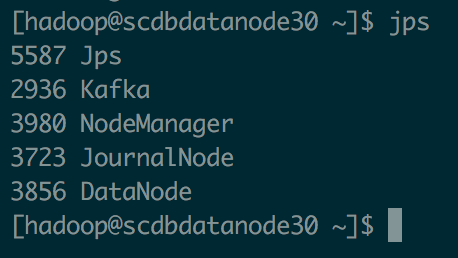
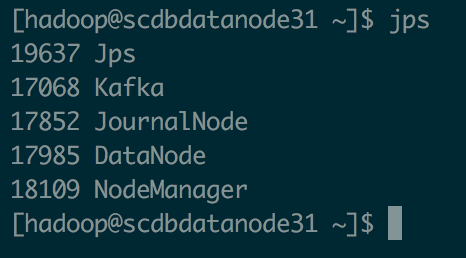
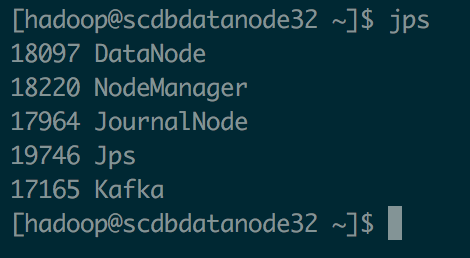
11.7、在RM（scdbdatanode27）所在master节点启动YARN:

#start-yarn.sh

11.8、在运行MRJS的scdbdatanode28上执行以下命令启动MR JobHistory Server：

#mr-jobhistory-daemon.sh start historyserver

至此，HDFS HA + YARN都成功启动完毕，在各个节点输入jps查看进程

 也可以用web查看：

HDFS HA界面：

http://133.37.31.80:50070/dfshealth.jsp

http://133.37.31.81:50070/dfshealth.jsp

YARN界面：

http://133.37.31.80:8088

## 12、停止集群

在RM和NN所在节点scdbdatanode27执行：

先停止yarn：

#stop-yarn.sh

再停止hdfs：

#stop-dfs.sh

在运行JobHistoryServer的scdbdatanode28上执行：

停止JobHistoryServer：

#mr-jobhistory-daemon.sh stop historyserver

## 13、再次启动

注意，再次启动时，所有的格式化命令都不用执行了！

## 14、执行MR示例程序

14.1、准备目录

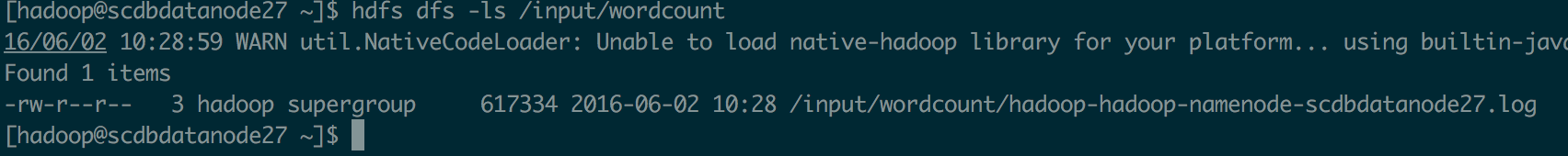
#hdfs dfs -mkdir /input

#hdfs dfs -mkdir /input/wordcount

#hdfs dfs -mkdir /output

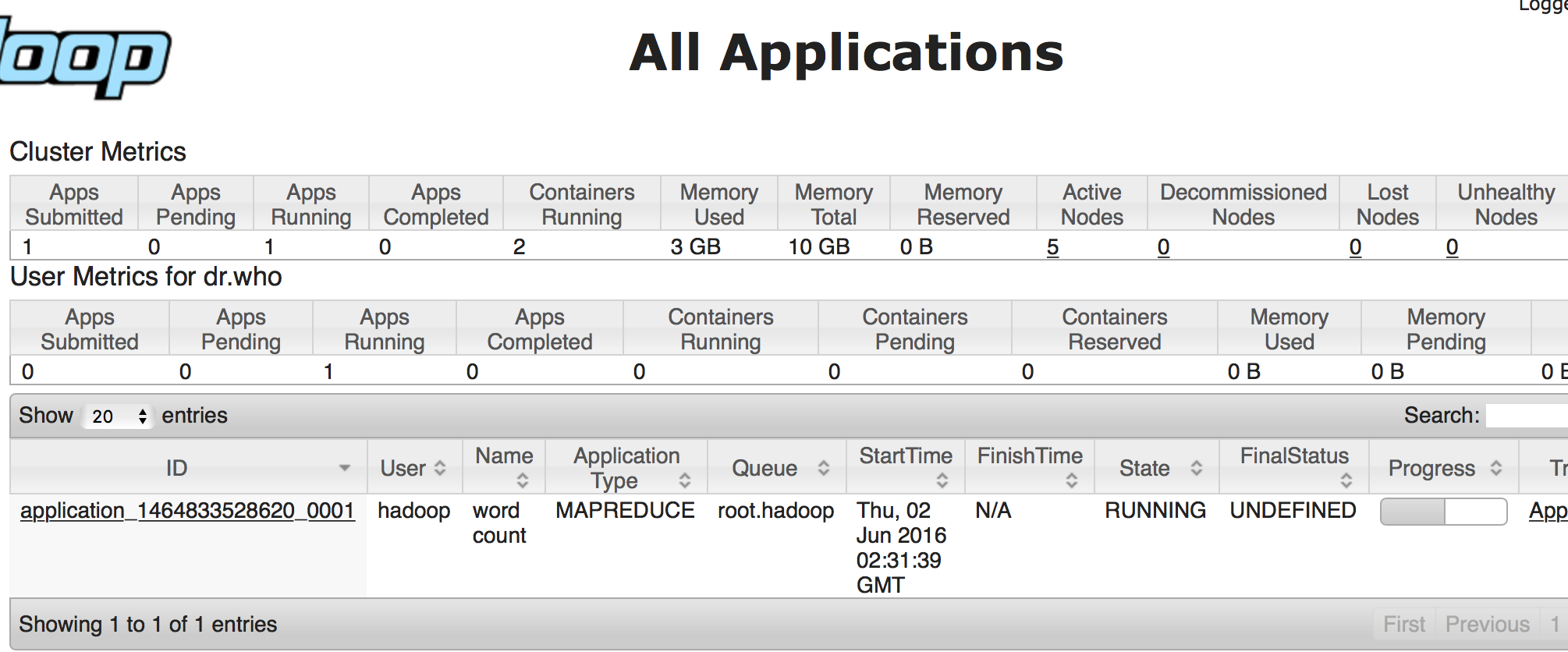
14.2、放入准备文件

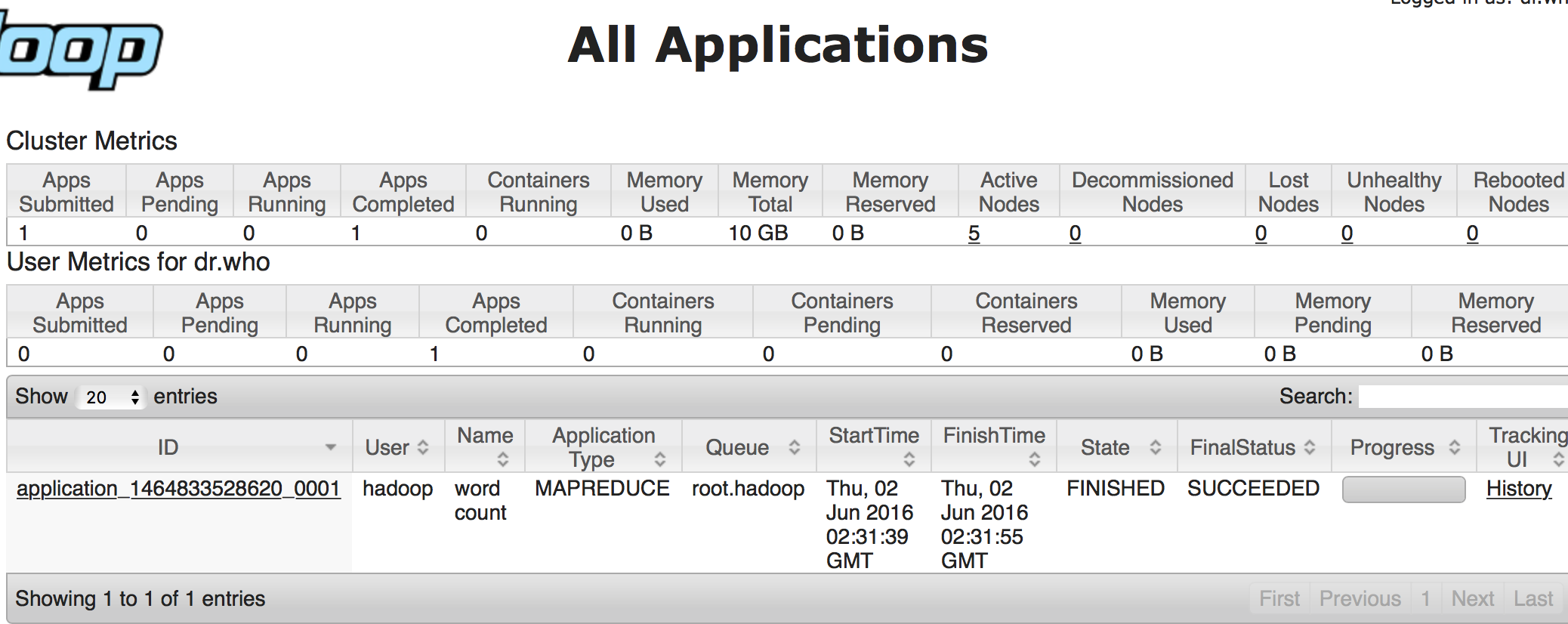
#hdfs dfs -put /hadoop/data1/logs/hadoop-hadoop-namenode-scdbdatanode27.log /input/wordcount

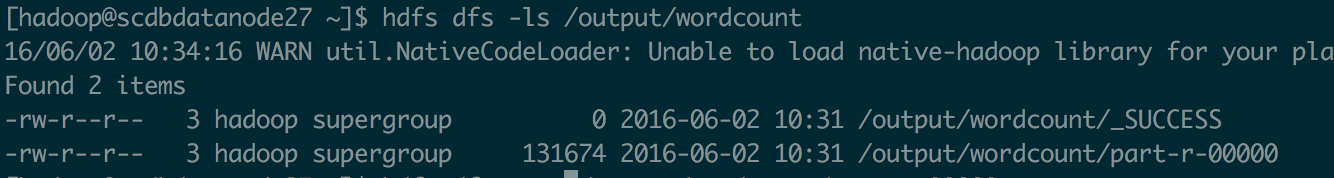


14.3、执行测试MR

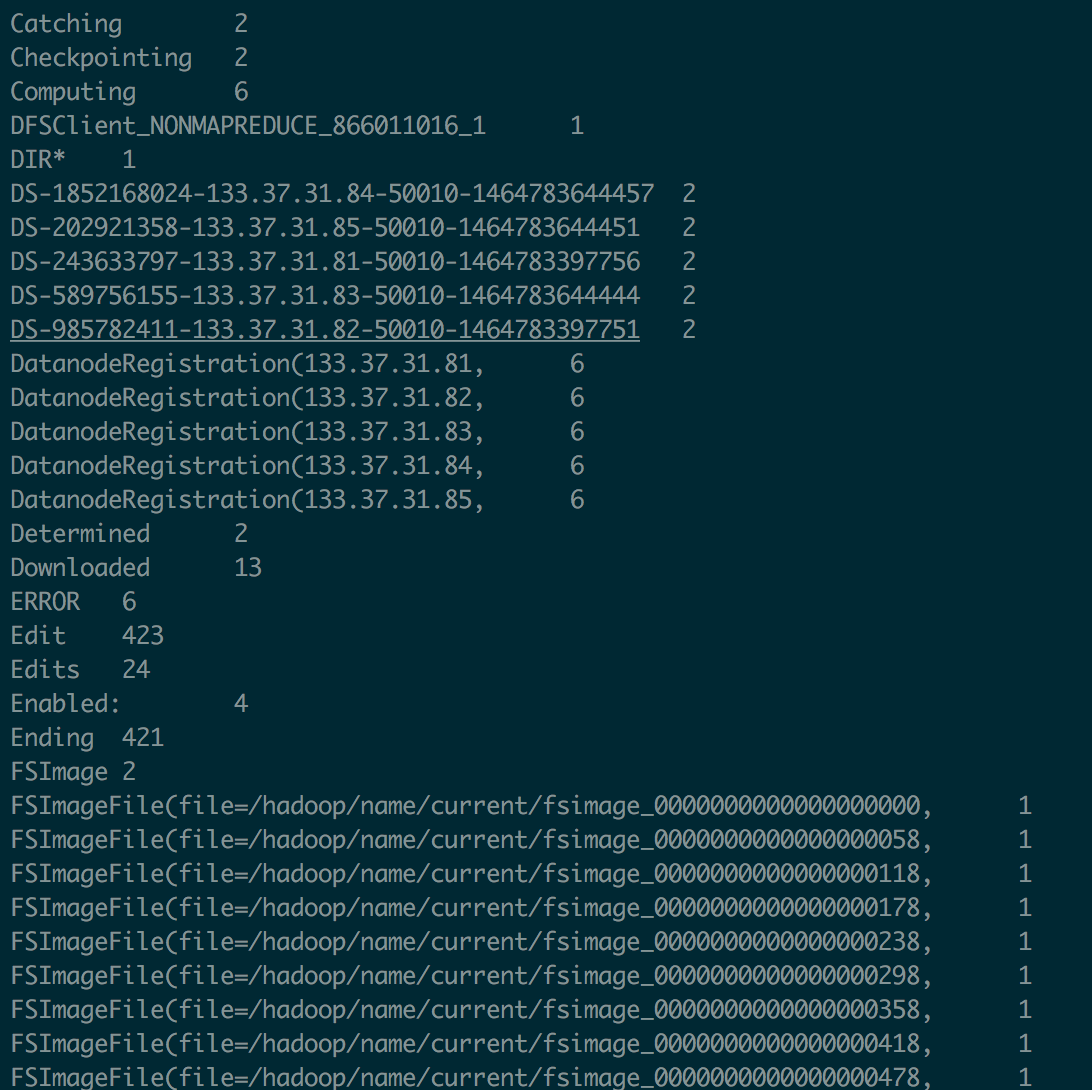
# hadoop jar /app/hadoop-2.2.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.2.0.jar wordcount /input/wordcount /output/wordcount



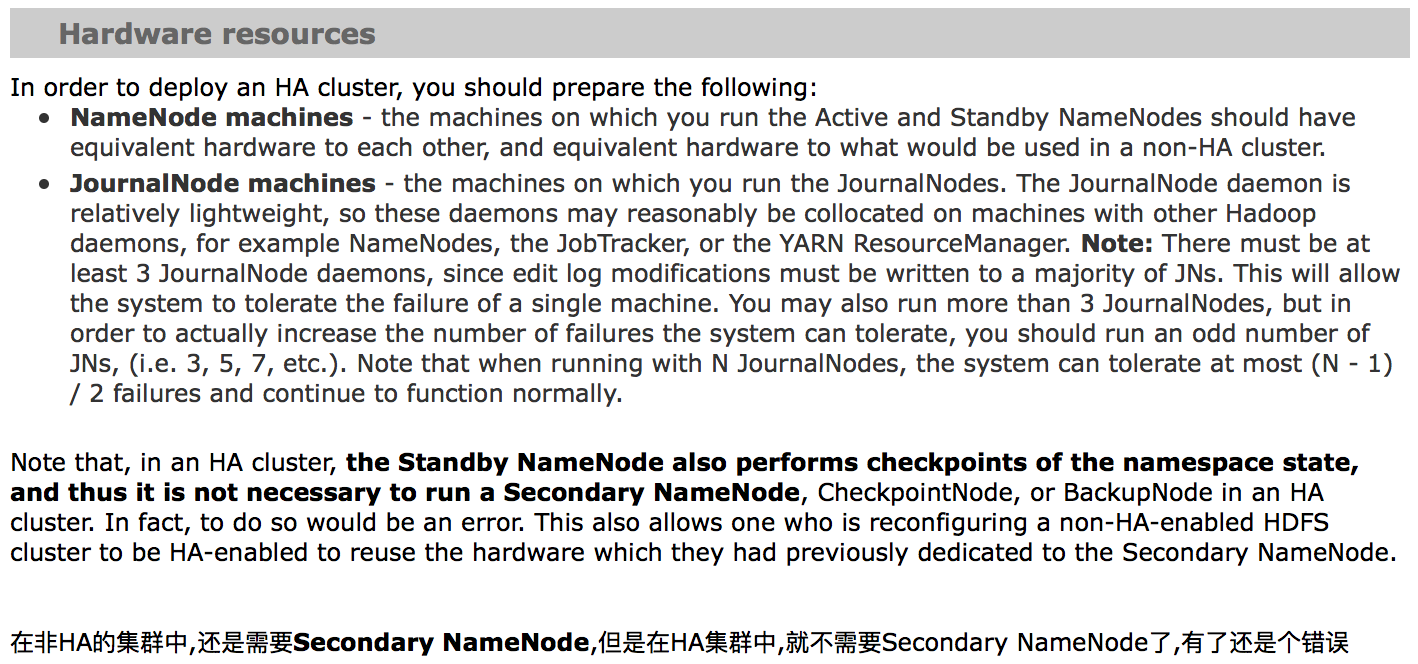




#hdfs dfs -cat /output/wordcount/part-r-00000



PS：JournalNode 和 Secondary NameNode存在关系



# 安装kafka

## 1、下载并解压

## 2、配置kafka：$KAFKA\_HOME/config/server.properties

修改或者新增以下内容

scdbdatanode27:

*broker.id=0*

*port=9092*

*host.name=scdbdatanode27*

*advertised.host.name=scdbdatanode27*

*log.dirs=/hadoop/data1/kafka-logs*

*num.partitions=1*

*zookeeper.connect=scdbdatanode27:2181,scdbdatanode28:2181,scdbdatanode29:2181*

scdbdatanode28:

*broker.id=1（根据机器顺序依次+1）*

*port=9092*

*host.name=scdbdatanode28（当前机器hostname）*

*advertised.host.name=scdbdatanode28（当前机器hostname）*

*log.dirs=/hadoop/data1/kafka-logs*

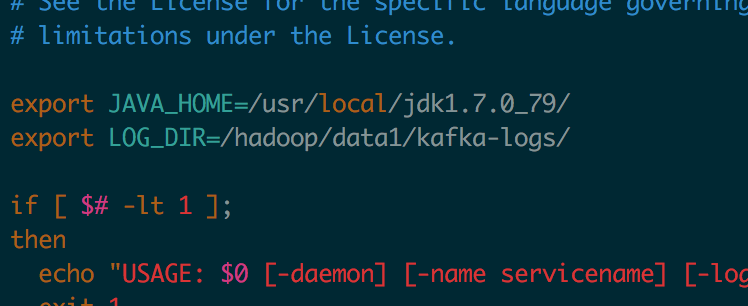
*num.partitions=1*

*zookeeper.connect=scdbdatanode27:2181,scdbdatanode28:2181,scdbdatanode29:2181（全部zk列表）*

## 3、修改$KAFKA\_HOME/bin/kafka-run-class.sh

*export JAVA\_HOME=/usr/local/jdk1.7.0\_79/*

*export LOG\_DIR=/hadoop/data1/kafka-logs/*



## 4、拷贝复制到所有计划安装kafka的节点，并修改$KAFKA\_HOME/config/server.properties

#scp -r /app/kafka\_2.11-0.9.0.1 scdbdatanode28:/app/

## 5、追加环境变量（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export KAFKA\_HOME=/app/kafka\_2.11-0.9.0.1" >> /etc/profile

#echo "export PATH=\$KAFKA\_HOME/bin:\$PATH" >> /etc/profile

 加载环境变量（hadoop用户）

#source /etc/profile

## 6、启动服务

#nohup kafka-server-start.sh /app/kafka\_2.11-0.9.0.1/config/server.properties &

## 7、验证

7.1、创建一个topic

#kafka-topics.sh --create --topic test --replication-factor 6 --partitions 1 --zookeeper scdbdatanode27:2181

计算机生成了可选文字:
Chadoop@scdbdatanode27 
.9.0.1]$ 
kafka-topics.sh --create 
Created topic "test" . 
Chadoop@scdbdatanode27 
.9.0. 
kafka 
kafka 
2 
2 
11- 
11- 
- -topic 
test 
repl i cation-factor 
6 --partitions 
1 zookeeper scdbdatanode27 : 2181 

7.2、假设我们在scdbdatanode28上，开一个终端，发送消息至kafka（scdbdatanode28模拟producer）

#kafka-console-producer.sh --broker-list scdbdatanode27:9092 --sync --topic test

在发送消息的终端输入：Hello Kafka

计算机生成了可选文字:
Chadoop@scdbdatanode28 
hellow 
test 
app] $ 
kafka-console-producer. sh 
--broker-list 
scdbdatanode27 
: 9092 
--sync 
--topic test 

7.3、假设我们在scdbdatanode29上，开一个终端，显示消息的消费（scdbdatanode29模拟consumer）

#kafka-console-consumer.sh --zookeeper scdbdatanode27:2181 --topic test --from-beginning

计算机生成了可选文字:
Chadoop@scdbdatanode29 
hellow 
test 
kafka-console-consumer 
. sh --zookeeper 
scdbdatanode27. 
•2181 - 
-topic test 
--from-beginning 

# 安装flume

参考： <http://my.oschina.net/cimu/blog/634961>

## 1、下载解压

## 2、追加环境变量

（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export FLUME\_HOME=/app/apache-flume-1.6.0-bin" >> /etc/profile

#echo "export FLUME\_CONF\_DIR=$FLUME\_HOME/conf" >> /etc/profile

#echo "export PATH=\$FLUME\_HOME/bin:\$PATH" >> /etc/profile

加载环境变量（hadoop用户）

#source /etc/profile

## 3、如果有必要修改JAVA版本，则在$FLUME\_HOME/conf下复制改名flume-env.sh.template为flume-env.sh

#cp flume-env.sh.template flume-env.sh

修改JAVA\_HOME和JAVA\_OPTS

#vi flume-env.sh

*exportJAVA\_HOME=/home/hadoop/jdk1.7.0\_75*

*exportJAVA\_OPTS="-Xms100m -Xmx2000m -Dcom.sun.management.jmxremote"*

## 4、在flume的conf文件夹下拷贝增加配置文件flume-conf.properties

#cp flume-conf.properties.template flume-conf.properties

计算机生成了可选文字:
agent 
agent. sinks 
# For 
agent 
agent 
agent 
# The 
agent 
= seqGenSrc 
. sources 
agent. channels = memoryChannel 
= loggerSink 
each one of the sources, 
. sources. 
. sources. 
. sources. 
channel 
. sources. 
# Each sink's 
the type is defined 
seqGenSrc. type 
= netcat 
seqGenSrc.bind = 
scdbdatanode27 
seqGenSrc.port = 
44444 
can be defined as follows. 
seqGenSrc. channels = memoryChannel 
type must be defined 
agent. sinks. loggerSink. type 
= logger 
#Specify the channel the sink should use 
agent. sinks. loggerSink. channel = memoryChannel 
# Each channel's type is defined. 
agent. channels. memoryChannel . type = memory 
# Other config values specific to each type of channel(sink or source) 
# can be defined as well 
# In this case, it specifies the capacity of the memory channel 
agent. channels. memoryChannel . capacit' = 100 

## 5、启动flume

#flume-ng agent -c /app/apache-flume-1.6.0-bin/conf/ -f /app/apache-flume-1.6.0-bin/conf/flume-conf.properties -n agent -Dflume.root.logger=INFO,console

其中"-n"选项的值表示Flume Agent的实例名称，"-f"选项的值表示配置文件的路径。

此时本机44444端口上的数据会通过该Agent被传递给日志处理框架，由于“-Dflume.root.logger=INFO,console”的设置，日志处理框架直接在控制台打印数据。

我们可通过telnet命令在44444端口上产生数据。

## 6、控制台验证flume

在任意终端执行命令

#telnet 133.37.31.80 44444

输入任意内容，在flume主机上会正确显示

输入：

计算机生成了可选文字:
hujundeMacBook-Pro:N hujun$ telnet 133.37.31.80 44444 
Trying 133.37.31.80.. 
Connected to 133.37.31.80. 
Escape character is 'A] ' 
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 
OK 
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb 
OK 

显示：

计算机生成了可选文字:
1 61 61 61 61 61 
2016-06-01 15:52. 
: 34,437 (SinkRunner-PollingRunner-DefaultSinkProcessor) C INFO - 
2016-06-01 15. 
• 52 
61 61 61 61 61 61 aaaaaaaaaaaaaaaa } 
•51, 790 (SinkRunner-PollingRunner-DefaultSinkProcessor) C INFO - 
2 62 62 62 62 62 62 62 62 62 62 62 62 bbbbbbbbbbbbbbbb } 
61 
org.apache.flume.sink 
org.apache.flume.sink 

## 7、编程验证flume待执行

1、工程pom.xml文件增加flume的依赖

2、配置log4j.properties

安装Hive

安装Presto

# 安装Hive

## 安装MySQL

### 1、准备环境

以下操作在root用户下

#groupadd mysql

#useradd -g mysql mysql

#mkdir /usr/local/mysql

#mkdir /hadoop/data1/mysql/data

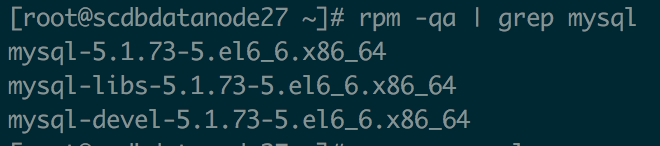
#chown -R mysql:mysql /usr/local/mysql

#chown -R hadoop:hadoop /hadoop/data1/mysql/

### 2、卸载老版本

使用下面的命令检查是否安装有MySQL Server

#rpm -qa | grep mysql



有的话通过下面的命令来卸载掉

rpm -e mysql //普通删除模式

rpm -e --nodeps mysql // 强力删除模式，如果使用上面命令删除时，提示有依赖的其它文件，则用该命令可以对其进行强力删除

#rpm -e mysql-devel-5.1.73-5.el6\_6.x86\_64

#rpm -e mysql-5.1.73-5.el6\_6.x86\_64

### 3、下载MySQL包mysql-5.6.25-linux-glibc2.5-x86\_64.tar.gz并解压

#tar xvf mysql-5.6.25-linux-glibc2.5-x86\_64.tar.gz

#cd /app/mysql-5.6.25-linux-glibc2.5-x86\_64

#mv /app/mysql-5.6.25-linux-glibc2.5-x86\_64/\* /usr/local/mysql/

### 4、初始化MySQL

# /usr/local/mysql/scripts/mysql\_install\_db --basedir=/usr/local/mysql --datadir=/hadoop/data1/mysql/data --user=mysql

### 5、创建服务启动命令

#cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld

### 6、修改数据目录

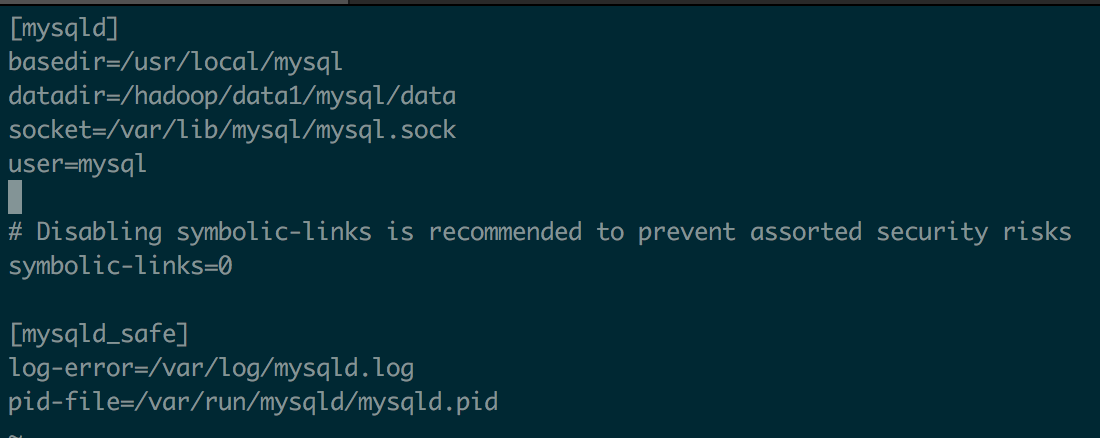
vi /etc/my.cnf

*basedir=/usr/local/mysql*

*datadir=/hadoop/data1/mysql/data*

*socket=/var/lib/mysql/mysql.sock*

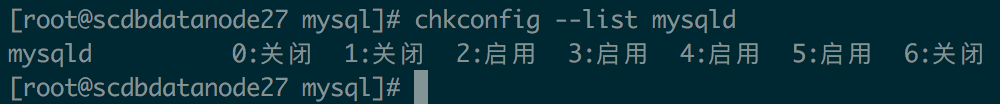
*user=mysql*



### 7、添加mysql到服务开机启动

# chkconfig mysqld on

# chkconfig --list mysqld



### 8、启动MySQL

#service mysqld start

### 9、添加PATH

#echo "export MYSQL\_HOME=/usr/local/mysql " >> /etc/profile

#echo "export PATH=\$MYSQL\_HOME/bin:\$PATH" >> /etc/profile

加载环境变量（mysql用户）

#source /etc/profile

### 10、执行 ln -s /var/lib/mysql/mysql.sock /tmp/mysql.sock

#ln -s /var/lib/mysql/mysql.sock /tmp/mysql.sock

### 11、修改root密码

进入MySQL

#mysqladmin -uroot

设置密码

#mysql> SET PASSWORD = PASSWORD('123456');

设置MYSQL的ROOT用户远程连接和密码

#mysql>use mysql;

#mysql>Grant all privileges on \*.\* to 'root'@'133.%' identified by 'mYsql&06' with grant option;

#mysql>flush privileges;

创建hadoop用户备用

#mysql>CREATE USER 'hadoop'@'133.%' IDENTIFIED BY 'OOPhad06';

创建Hive数据库

#mysql>create database hive DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

授权hadoop用户访问Hive数据库

#mysql>grant all privileges on hive.\* to hadoop;

#mysql>flush privileges;

其他

#service mysqld start

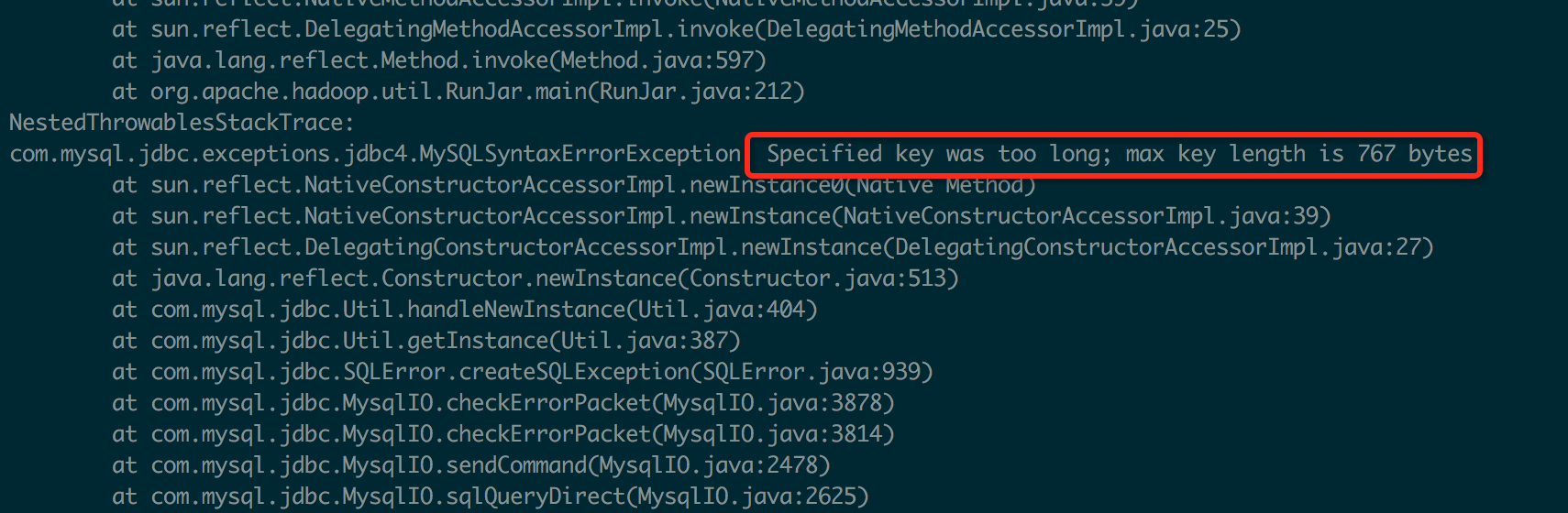
#service mysqld stop

#service mysqld restart

数据库其他要求

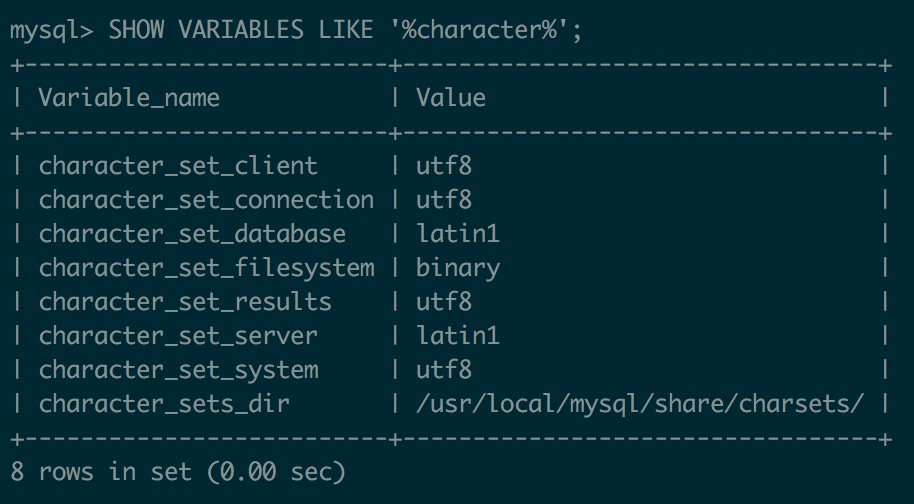
1、修改编码

针对以下异常情况，需要修改数据库编码：



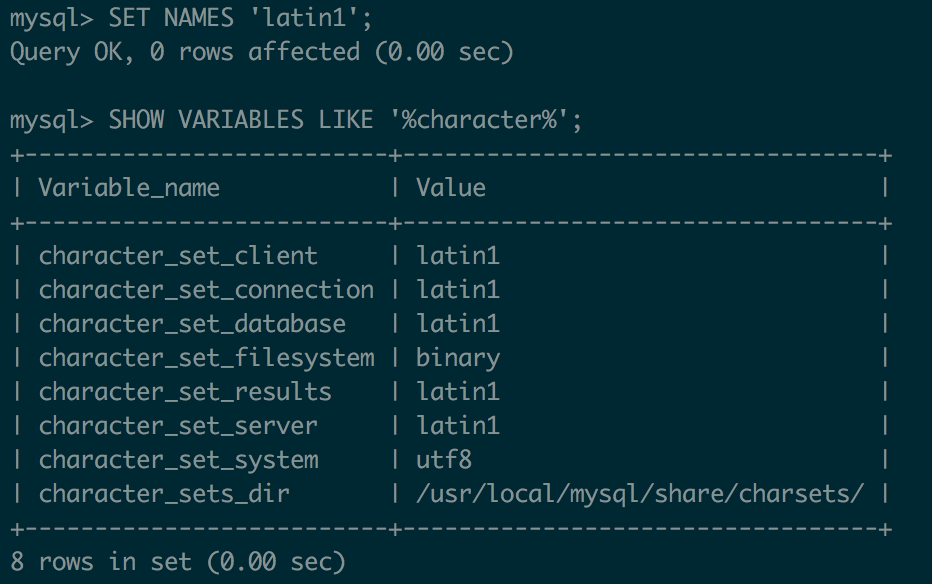
执行命令查看当前编码：

#mysql> SHOW VARIABLES LIKE '%character%';



执行命令修改编码：

mysql> SET NAMES 'latin1';



如果是出现异常再修改的编码，则需要删除数据库中已经有的表。并且重启。

最后关键的方法解决！！！【alter database hive character set latin1; 】

2、数据库引擎

查看当前数据库引擎，最好使用InnoDB模式

mysql> show engines;

3、修改数据库的binlog

mysql> set global binlog\_format='MIXED';

READ-COMMITTED需要把bin-log以mixed方式来记录  否则进入hive，会如下错误  FAILED: Error in metadata: javax.jdo.JDOException: Couldnt obtain a new sequence (unique id) : Binary logging not possible. Message: Transaction level 'READ-COMMITTED' in InnoDB is not safe for binlog mode 'STATEMENT'

## 安装Hive

### 1、下载解压

#tar -xzvf /app/hive-0.11.0.tar.gz -C /app/

### 2、追加环境变量

（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export HIVE\_HOME=/app/hive-0.11.0" >> /etc/profile

#echo "export PATH=\$HIVE\_HOME/bin:\$PATH" >> /etc/profile

加载环境变量（hadoop用户）

#source /etc/profile

### 3、准备mysql连接jar文件

#cp mysql-connector-java-5.1.38-bin.jar /app/hive-0.11.0/lib/

### 4、修改配置文件

#cp /app/hive-0.11.0/conf/hive-default.xml.template /app/hive-0.11.0/conf/hive-site.xml

#vi /app/hive-0.11.0/conf/hive-site.xml

*<property>*

*<name>hive.exec.scratchdir</name>*

*<value>/hadoop/data1/logs/hive</value>*

*<description>Scratch space for Hive jobs</description>*

*</property>*

*<property>*

*<name>hive.exec.local.scratchdir</name>*

*<value>/hadoop/data1/logs/hive/tmp/${user.name}</value>*

*<description>Local scratch space for Hive jobs</description>*

*</property>*

*<property>*

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

*<value>jdbc:mysql://scdbdatanode27:3306/hive?createDatabaseIfNotExist=true&amp;useUnicode=true&amp;characterEncoding=latin1&amp;autoReconnect=true&amp;failOverReadOnly=false</value>*

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

*</property>*

*<property>*

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

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

*</property>*

*<property>*

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

*<value>hadoop</value>*

*</property>*

*<property>*

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

*<value>OOPhad06</value>*

*</property>*

### 5、运行hive

一般启动方法

#hive

#hive --service cli（不清楚作用）

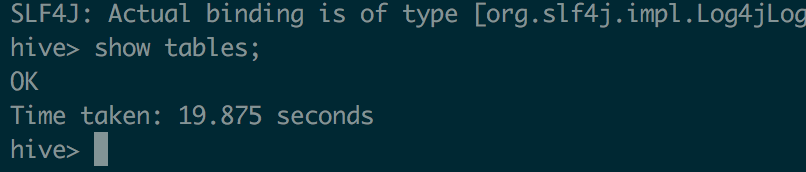
#hive --service hwi（不清楚作用）

查看日志启动方法

#hive -hiveconf hive.root.logger=DEBUG,console



hive> show tables;



# 安装Presto

参考文档： <http://www.tuicool.com/articles/vU7rYzR>

## 1、下载解压

#tar -zxvf /app/presto-server-0.147.tar.gz -C /app

## 2、准备配置文件

#mkdir /app/presto-server-0.147/etc

在该目录下存在以下文件：

node.properties ：每个节点的环境配置

jvm.config ：jvm 参数

config.properties ：配置 Presto Server 参数

log.properties ：配置日志等级

Catalog Properties ：Catalog 的配置

## 3、配置config.properties

#vi /app/presto-server-0.147/etc/config.properties

*node-scheduler.include-coordinator=false*

*http-server.http.port=50031*

*query.max-memory=50GB*

*#query.max-memory-per-node=5GB*

*discovery-server.enabled=true*

*discovery.uri=http://133.37.31.80:50031*

## 4、配置jvm.config

#vi /app/presto-server-0.147/etc/jvm.config

*-server*

*-Xmx16G*

*-XX:+UseG1GC*

*-XX:G1HeapRegionSize=32M*

*-XX:+UseGCOverheadLimit*

*-XX:+ExplicitGCInvokesConcurrent*

*-XX:+HeapDumpOnOutOfMemoryError*

*-XX:OnOutOfMemoryError=kill -9 %p*

## 5、配置log.properties

#vi /app/presto-server-0.147/etc/log.properties

*com.facebook.presto=INFO*

## 6、配置node.properties

需要提前创建好目录：/hadoop/data1/presto

#vi /app/presto-server-0.147/etc/node.properties

*node.environment=tydic\_dpi\_presto*

*node.id=tydic\_dpi\_scdbdatanode27*

*node.data-dir=/hadoop/data1/presto/data*

## 7、追加环境变量

（需要试用root用户追加，全局生效），需要在所有节点执行

#echo "export PRESTO\_HOME=/app/presto-server-0.147" >> /etc/profile

#echo "export PATH=\$PRESTO\_HOME/bin:\$PATH" >> /etc/profile

加载环境变量（hadoop用户）

#source /etc/profile

## 8、启动

### 8.1、设置临时1.8JDK

export PATH=/usr/local/jdk1.8.0\_91/bin:$PATH

为了不影响其他大数据环境程序运行，且临时参数关闭shell就会失效，建议新开一个shell来临时设置并运行Presto。

### 8.2、启动方式

后台启动

#/app/presto-server-0.147/bin/launcher start

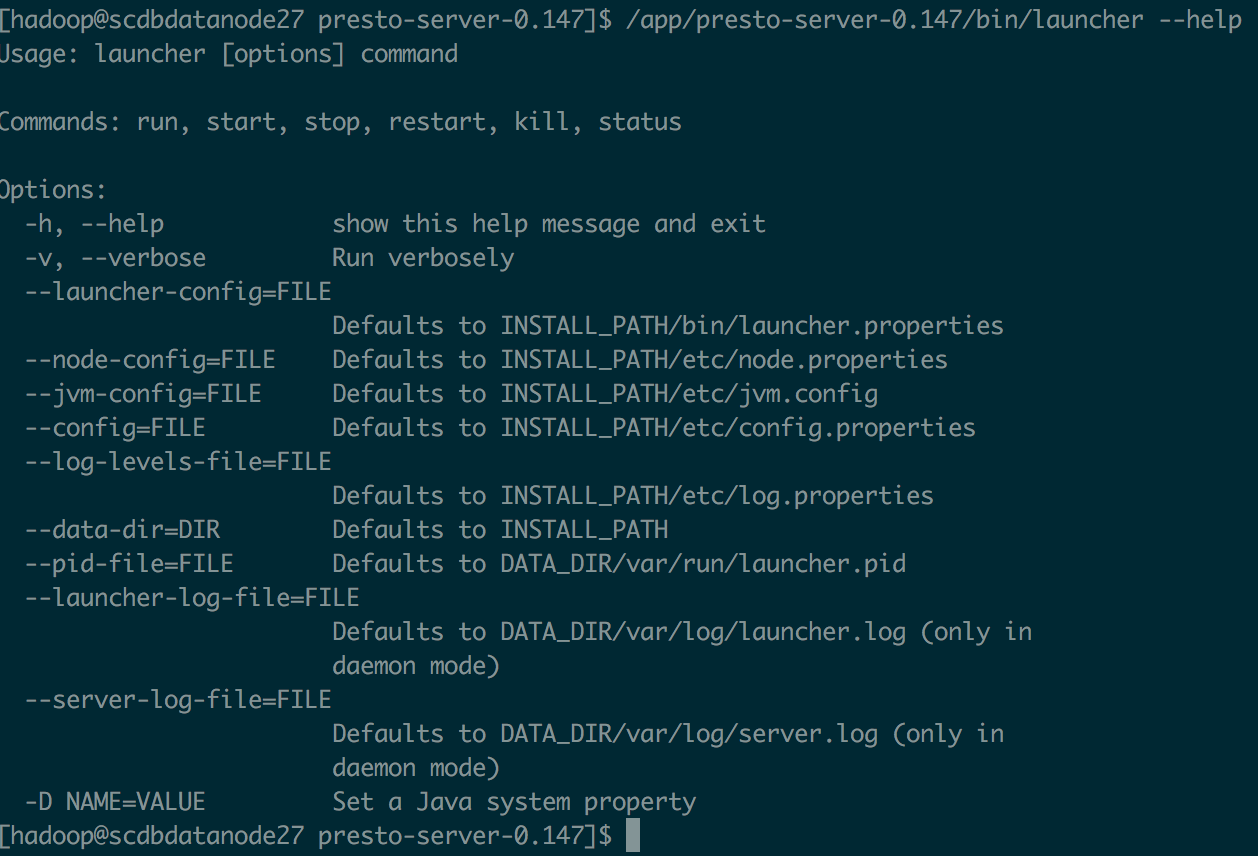
前台启动

#/app/presto-server-0.147/bin/launcher run

停止

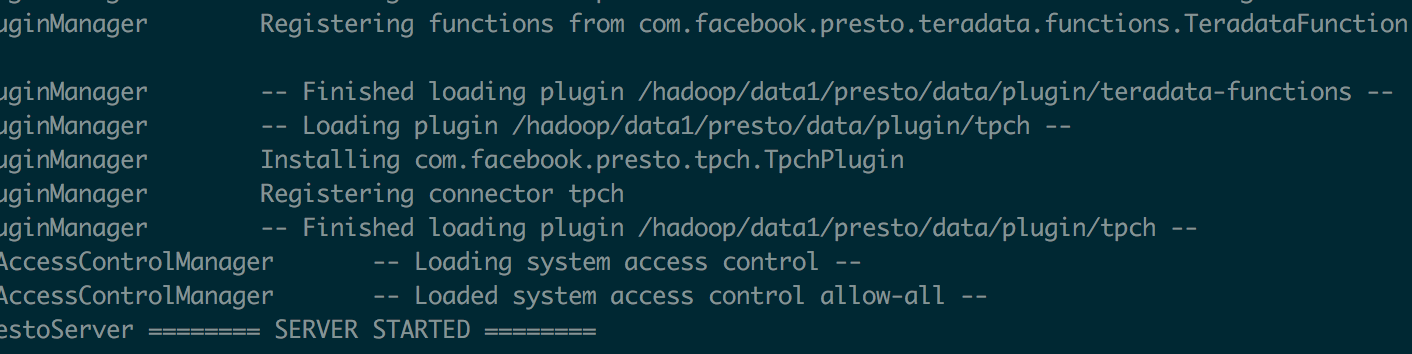
#/app/presto-server-0.147/bin/launcher stop

更多启动参数



## 9、查看日志

tail -fn 1000 /hadoop/data1/presto/data/var/log/server.log

**