1. **单机部署**
2. **环境准备**

1.磁盘挂载之后，赋予所有data目录777权限

2.操作系统环境以centos7.5为例，节点规划如下：

|  |  |
| --- | --- |
| 节点名 | 部署进程 |
| 192.168.10.1 | NameNode、DataNode、SecondaryNameNode、ResourceManager、NodeManager、QuorumPeerMain、Kafka、statestored、catalogd、impalad |

* 1. **修改操作系统参数**
* 为**每台**机器关闭防火墙并设置seleniux策略；root用户操作

[root@rhino001]# systemctl stop firewalld.service

[root@rhino001]# systemctl disable firewalld.service

[root@rhino001]# setenforce 0

[root@rhino001]# vim /etc/selinux/config

|  |
| --- |
| # This file controls the state of SELinux on the system.  # SELINUX= can take one of these three values:  # enforcing - SELinux security policy is enforced.  # permissive - SELinux prints warnings instead of enforcing.  # disabled - No SELinux policy is loaded.  SELINUX=disabled  # SELINUXTYPE= can take one of three two values:  # targeted - Targeted processes are protected,  # minimum - Modification of targeted policy. Only selected processes are protected.  # mls - Multi Level Security protection.  SELINUXTYPE=targeted |

* 修改打开的文件句柄数

[root@rhino001]# vim /etc/security/limits.conf

在文件最下方添加如下内容：

|  |
| --- |
| \* soft nofile 65536  \* hard nofile 65536  \* soft nproc 65536  \* hard nproc 65536  \* soft memlock -1  \* hard memlock -1 |

[root@rhino001]# vim /etc/security/limits.d/20-nproc.conf

修改如下内容

|  |
| --- |
| \* soft nproc 65536 |

[root@rhino001]#vi /etc/pam.d/config-util

在文件最后加入一行

|  |
| --- |
| session required pam\_limits.so |

[root@rhino001]#vi /etc/profile

添加如下两行

ulimit -SHn 65536

ulimit -SHu 65536

[root@rhino001]#vi /etc/sysctl.conf (该配置生效需要重启)

|  |
| --- |
| vm.max\_map\_count=262144 |

也可以通过sysctl -w vm.max\_map\_count=262144使得该配置临时生效。

* 重启系统使各项配置生效

[root@rhino001]# reboot

* 1. **配置rhino用户**
* 新建rhino用户并为它设定密码。

[root@rhino001 ~]# useradd -g root -d /home/rhino -m rhino

[root@rhino001 ~]# passwd rhino

* 以rhino用户将相关软件包拷贝到rhino用户/home/rhino目录下
  1. **设置hosts**

以root用户登录，配置主机名称

[root@rhino001 ~]# vim /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost  192.168.10.1 rhino001 |

**注意**：首行必须配置为“127.0.0.1 localhost”，不能使用其他配置。

* 1. **创建单机rhino用户的无密码访问**
* 以rhino用户登录，创建.ssh目录(如果目录存在则忽略此步)

[rhino@rhino001 ~]$ mkdir ~/.ssh

* 输入命令：ssh-keygen -t rsa生成public key(id\_rsa.pub)和private key(id\_rsa)文件。当出现”Enter Passphrase”的提示时直接按回车

[rhino@rhino001 ~]$ ssh-keygen -t rsa

* 将id\_dsa.pub的内容添加到~/.ssh/authorized\_keys中

[rhino@rhino001 ~]$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

* 修改目录文件权限

[rhino@rhino001 ~]$ chmod 700 ~/.ssh

[rhino@rhino001 ~]$ chmod 600 ~/.ssh/authorized\_keys

* 测试是否不需要密码即可登陆

[rhino@rhino001 ~]$ ssh rhino001

* 1. **安装JDK**
* 以rhino用户安装JDK，安装包名为jdk1.8.0\_181.tar.gz

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar zxvf jdk1.8.0\_181.tar.gz

* 配置JAVA\_HOME环境变量

[rhino@rhino001 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export PATH=$PATH:$JAVA\_HOME/bin: |

* 使环境变量生效

[rhino@rhino001 ~]$ source ~/.bashrc

1. **安装MYSQL**
   1. **安装部署**

* 以rhino用户执行如下安装命令

[rhino@rhino001 ~]$cd /home/rhino

[rhino@rhino001 ~]$tar -zxvf mysql-5.6.25-linux-x86\_64.tar.gz

[rhino@rhino001~]$mkdir -p ~/mysql-5.6.25-linux-x86\_64/mydata/lock/subsys

[rhino@rhino001~]$mkdir -p ~/mysql-5.6.25-linux-x86\_64/mydata/tmp

* 修改环境变量

[rhino@rhino001 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export LD\_LIBRARY\_PATH=/home/rhino/mysql-5.6.25-linux-x86\_64/lib:$LD\_LIBRARY\_PATH  export PATH=$JAVA\_HOME/bin:$PATH |

[rhino@rhino001 ~]$ source ~/.bashrc

* 配置my.cnf

[rhino@rhino001 ~]$ vim ~/mysql-5.6.25-linux-x86\_64/my.cnf

|  |
| --- |
| [client]  port=3306  default-character-set=utf8    [mysqld]  join\_buffer\_size=16M  sort\_buffer\_size=16M  read\_rnd\_buffer\_size=16M  max\_allowed\_packet=200M  max\_connections=1000  port=3306  socket= /home/rhino/mysql-5.6.25-linux-x86\_64/mydata/tmp/mysql.sock  character-set-server=latin1  basedir=/home/rhino/mysql-5.6.25-linux-x86\_64  datadir=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata  explicit\_defaults\_for\_timestamp=true  binlog\_format=MIXED  log-bin=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata/master-bin  server\_id=131  auto\_increment\_increment=1  auto\_increment\_offset=1  #binlog-do-db=test  #binlog-ignore-db=mysql  #replicate-do-db=test  #replicate-ignore-db=mysql  log-slave-updates  slave-skip-errors=all  sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES |

* 配置mysql.server

[rhino@rhino001 ~]$ vim ~/mysql-5.6.25-linux-x86\_64/support-files/mysql.server

在第一次出现basedir、datadir及lockdir的地方修改其值。

|  |
| --- |
| basedir=/home/rhino/mysql-5.6.25-linux-x86\_64/  datadir=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata  lockdir='/home/rhino/mysql-5.6.25-linux-x86\_64/mydata/lock/subsys' |
| * 切换至root用户，注释掉mysql的系统安装时的配置文件   [root@rhino001 ~]$ cd /etc  [root@rhino001 etc]$ mv my.cnf my.cnf.bak |

* 1. **初始化及启动mysql数据库**
* 初始化数据库

[rhino@rhino001 ~]$ cd /home/rhino/mysql-5.6.25-linux-x86\_64

[rhino@rhino001~]$scripts/mysql\_install\_db --defaults-file=/home/rhino/mysql-5.6.25-linux-x86\_64/my.cnf --datadir=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata --basedir=/home/rhino/mysql-5.6.25-linux-x86\_64

* 机器上启动mysql

[rhino@rhino001 ~]$ cd mysql-5.6.25-linux-x86\_64/

[rhino@rhino001~]$ nohup bin/mysqld\_safe --defaults-file=/home/rhino/mysql-5.6.25-linux-x86\_64/my.cnf --user=rhino &

* 1. **添加rhino用户**
* 登陆mysql并为其创建rhino用户、登录密码及设置权限

**注：此处的用户名及密码应与spark配置文件hive-site.xml中配置的相同**。

[rhino@rhino001 ~]$ cd mysql-5.6.25-linux-x86\_64/

[rhino@rhino001~]$bin/mysql --socket=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata/tmp/mysql.sock -P3306 -uroot

* 登录成功，执行以下操作

mysql> CREATE USER 'rhino'@'%' IDENTIFIED BY 'rhino';

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'rhino'@'%' WITH GRANT OPTION;

mysql> delete from mysql.user where user='';

mysql> flush privileges;

* 修改最大连接数：

mysql> show variables like 'max\_connections';

+-----------------+-------+---------

| Variable\_name | Value |

+-----------------+-------+----------

| max\_connections | 151 |

+-----------------+-------+----------

1 row in set (0.01 sec)

mysql> set global max\_connections=1000;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

* 验证能否用新建的rhino用户登陆

[rhino@rhino001 ~]$ /home/rhino/mysql-5.6.25-linux-x86\_64/bin/mysql --socket=/home/rhino/mysql-5.6.25-linux-x86\_64/mydata/tmp/mysql.sock -P3306 -urhino -prhino

* 添加mysql开机自启

切换至root用户

[root@rhino001 ~]# vim /etc/rc.d/rc.local

添加以下命令，并保存

|  |
| --- |
| su - rhino -c "cd /home/rhino/mysql-5.6.25-linux-x86\_64; bin/mysqld\_safe --defaults-file=/home/rhino/mysql-5.6.25-linux-x86\_64/my.cnf --user=rhino &" |

* 1. **注意点（如果安装两台mysql）**

**注：**另一个数据库的安装步骤同上，注意点如下：



1. **保证port端口没有被占用，不要与前一个mysql重复；**
2. **Character-set-server修改为utf-8(cloud库使用utf-8;shark库和hive\_rhino库使用latin1)**
3. **Basedir、datadir、log-bin的绝对路径，需要根据mysql的实际路径进行修改;（前一个mysql安装时也一样）**

假设第一台数据库默认安装前台库，端口是3306，编码为utf8；第二台数据库安装后台库，端口是3307，编码是latin1。

* 后台库配置文件参考如下：

[rhino@rhino001 ~]$ vim ~/mysql-5.6.25-linux-x86\_64-shark /my.cnf

|  |
| --- |
| [client]  port=3307  default-icharacter-set=latin1  [mysqld]  join\_buffer\_size=16M  sort\_buffer\_size=16M  read\_rnd\_buffer\_size=16M  port=3307  socket=/home/rhino/mysql-5.6.25-shark/mydata/tmp/mysql.sock  character-set-server=latin1  #character-set-server=utf8  basedir=/home/rhino/mysql-5.6.25-shark/  datadir=/home/rhino/mysql-5.6.25-shark/mydata  explicit\_defaults\_for\_timestamp=true  binlog\_format=MIXED  log-bin=/home/rhino/mysql-5.6.25-shark/mydata/master-bin  server\_id=132  auto\_increment\_increment=1  auto\_increment\_offset=1  #binlog-do-db=test  #binlog-ignore-db=mysql  #replicate-do-db=test  #replicate-ignore-db=mysql  log-slave-updates  slave-skip-errors=all  sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES |

1. **部署HDFS**
   1. **安装Zookeeper**

* 解压zookeeper-3.4.5-cdh5.7.0.tar.gz

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf zookeeper-3.4.5-cdh5.7.0.tar.gz

* 修改zookeeper配置文件

[rhino@rhino001 ~]$ cd ~/zookeeper-3.4.5-cdh5.7.0/conf/

[rhino@rhino001 ~]$ mv zoo\_sample.cfg zoo.cfg

[rhino@rhino001 ~]$ vim zoo.cfg

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/home/rhino/zookeeper-3.4.5-cdh5.7.0/data  clientPort=2181  #  # 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  #autopurge.purgeInterval=1  server.1=rhino212:2888:3888 |

* 创建数据目录并创建myid文件

[rhino@rhino001 ~]$ mkdir /home/rhino/zookeeper-3.4.5-cdh5.7.0/data

[rhino@rhino001 ~]$ cd /home/rhino/zookeeper-3.4.5-cdh5.7.0/data

[rhino@rhino001 ~]$ touch myid; echo 1 > myid;

* 修改环境变量

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export LD\_LIBRARY\_PATH=/home/rhino/mysql-5.6.25-linux-x86\_64/lib:$LD\_LIBRARY\_PATH  export ZOOKEEPER\_HOME=/home/rhino/zookeeper-3.4.5-cdh5.7.0  export PATH=$JAVA\_HOME/bin:$ZOOKEEPER\_HOME/bin:$PATH |

* 使环境变量生效

[rhino@rhino001 ~]$ source ~/.bashrc

* 启动zookeeper

[rhino@rhino001 ~]$ zkServer.sh start

* 查看zookeeper状态

[rhino@rhino001 ~]$ zkServer.sh status

显示standalone则表示zookeeper启动成功。

* zookeeper停止命令（需要时停止）

[rhino@rhino001 ~]$ zkServer.sh stop

* 1. **安装HDFS**
* 解压Hadoop安装包(hadoop-2.6.0-cdh5.7.0.tar.gz)

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf hadoop-2.6.0-cdh5.7.0.tar.gz

* + 1. **配置slaves节点**

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/

[rhino@rhino001 hadoop]$ vim slaves

|  |
| --- |
| rhino001 #注：slave的主机名 |

* + 1. **配置core-site.xml**

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/

[rhino@rhino001 hadoop]$ vim core-site.xml

|  |
| --- |
| <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://rhino001:9020</value>  </property>  <property>  <name>ha.zookeeper.quorum</name>  <value>rhino001:2181</value>  </property>  <property>>  <name>hadoop.tmp.dir</name>  <value>file:/home/rhino/tmp</value>  </property>  <property>  <name>fs.trash.interval</name>  <value>60</value>  </property>  <property>  <name>io.file.buffer.size</name>  <value>131072</value>  </property>  </configuration> |

* + 1. **配置hdfs-site.xml**

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/

[rhino@rhino001 hadoop]$ vim hdfs-site.xml

**全文复制以下内容，并做相应的修改。**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>dfs.client.read.shortcircuit</name>  <value>true</value>  </property>  <property>  <name>dfs.domain.socket.path</name>  <value>/home/rhino/hdfs-sockets/dn</value>  </property>  <property>  <name>dfs.datanode.hdfs-blocks-metadata.enabled</name>  <value>true</value>  </property>  <property>  <name>dfs.client.file-block-storage-locations.timeout.millis</name>  <value>10000</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///home/rhino/namenode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>**/data01**/datanode,**/data02**/datanode,/data03/datanode,/data04/datanode,/data05/datanode,/data06/datanode </value>  </property>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.permissions.enabled</name>  <value>false</value>  </property>  <property>  <name>fs.hdfs.impl.disable.cache</name>  <value>true</value>  </property>  </configuration> |

* + 1. **安装及初始化**
* 配置HADOOP\_HOME环境变量

[rhino@rhino001 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export LD\_LIBRARY\_PATH=/home/rhino/mysql-5.6.25-linux-x86\_64/lib:$LD\_LIBRARY\_PATH  export ZOOKEEPER\_HOME=/home/rhino/zookeeper-3.4.5-cdh5.7.0  export HADOOP\_HOME=/home/rhino/hadoop-2.6.0-cdh5.7.0  export HADOOP\_PID\_DIR=/home/rhino/pids  export YARN\_PID\_DIR=/home/rhino/pids  export PATH=$JAVA\_HOME/bin:$PATH:$ZOOKEEPER\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin |

* 使环境变量生效

[rhino@rhino001 ~]$ source ~/.bashrc

* 创建hdfs-sockets目录

[rhino@rhino001 ~]$ mkdir /home/rhino/hdfs-sockets

* 修改hdfs-sockets组权限

切换到root用户执行以下命令

[root@rhino001 ~]# chown rhino:root /home/rhino/hdfs-sockets

* 以rhino用户进行namenode格式化

[rhino@rhino001 ~]$ hdfs namenode -format

* 启动HDFS

[rhino@rhino001 ~]$ start-dfs.sh

* 检查HDFS是否正常启动

使用浏览器登录到hdfs监控页面http://ip:50070/查看Live Nodes是否为1(即配置的slaves节点个数)，如果为1即说明hdfs已正常启动。



* 停止HDFS(需要停止时执行)

运行如下命令：

[rhino@rhino001 ~]$ stop-dfs.sh

1. **部署impala集群**
   1. **安装impala**

* 以rhino用户在rhino001机器下解压impala安装包impala-2.12.0.tar.gz

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf impala-2.12.0.tar.gz

[rhino@rhino001 ~]$ ln -s impala-2.12.0/ impala

* 拷贝/mysql-connector-java-5.1.31.jar包

cp /home/rhino/impala/lib/mysql/mysql-connector-java-5.1.31.jar /home/rhino/impala/lib/impala/lib/

* 拷贝**hadoop配置文件**到impala配置中

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop

[rhino@rhino001 hadoop]$ cp core-site.xml hdfs-site.xml /home/rhino/impala/lib/impala/conf

* 配置**hive-site.xml**配置文件

[rhino@rhino001 ~]$ vim /home/rhino/impala/lib/impala/conf/hive-site.xml

|  |
| --- |
| <configuration>  <property>  <name>hive.metastore.client.socket.timeout</name>  <value>3600</value>  <description>MetaStore Client socket timeout in seconds</description>  </property>  <property>  <name>javax.jdo.option.ConnectionURL</name> <value>jdbc:mysql://ip:3306/hive\_rhino?createDatabaseIfNotExist=true</value>  </property>  <property>  <name>javax.jdo.option.ConnectionDriverName</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>javax.jdo.option.ConnectionUserName</name>  <value>rhino</value>  </property>  <property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>rhino</value>  </property>  <property>  <name>hive.querylog.location</name>  <value>${user.home}/hive-logs/querylog</value>  </property>  </configuration> |

* 配置**impala-env.sh**配置文件

[rhino@rhino001 ~]$ vim /home/rhino/impala/sbin/impala-env.sh

|  |
| --- |
| export IMPALA\_HOME=~/impala/lib/impala  export IMPALA\_BIN=$IMPALA\_HOME/sbin  export IMPALA\_CONF\_DIR=$IMPALA\_HOME/conf  export LIBHDFS\_OPTS=$IMPALA\_HOME/lib  export MYSQL\_CONNECTOR\_JAR=${MYSQL\_CONNECTOR\_JAR:$IMPALA\_HOME/lib/mysql-connector-java-5.1.31.jar}  export IMPALA\_SHELL\_HOME=$IMPALA\_HOME/../impala-shell  #**java tool options,具体-Xmx内存大小需要根据实际情况确定**  export IMPALAD\_JAVA\_TOOL\_OPTIONS=""  export CATALOGD\_JAVA\_TOOL\_OPTIONS="-Xmx**20G** -XX:+UseG1GC -XX:+UnlockExperimentalVMOptions -XX:MaxGCPauseMillis=200 -XX:G1NewSizePercent=1 -XX:InitiatingHeapOccupancyPercent=65 -XX:+ParallelRefProcEnabled -XX:ConcGCThreads=4 -XX:ParallelGCThreads=16 -XX:MaxTenuringThreshold=1 -XX:G1HeapRegionSize=32m -XX:G1MixedGCCountTarget=32 -XX:G1OldCSetRegionThresholdPercent=2 -XX:SurvivorRatio=4 -XX:+UnlockDiagnosticVMOptions -XX:+G1SummarizeConcMark"  #**impalad args,需要在对应路径下，新建下面的目录**  export IMPALA\_ARGS="--scratch\_dirs**=/data01/impala\_tmp,/data02/impala\_tmp,/data03/impala\_tmp,/data04/impala\_tmp,/data05/impala\_tmp,/data06/impala\_tmp**"  #catalogd args  export CATALOGD\_ARGS="--num\_metadata\_loading\_threads=12 "  #statestore args  export STATESTORE\_ARGS=""  export STATE\_STORE\_HOST=**"192.168.10. 1"**  export CATALOG\_SERVICE\_HOST**="192.168.10.1"**  export impalad\_ips=( **"rhino001"**); |

* 1. **启动impala各进程**
* 在rhino001机器上启动statestored，catalogd，impalad进程

[rhino@rhino001 ~]$ ~/impala/sbin/start-statestored.sh

[rhino@rhino001 ~]$ ~/impala/sbin/start-catalogd.sh

[rhino@rhino001~]$

~/impala/sbin/start-impalad.sh -state\_store\_host=192.168.10.1 -catalog\_service\_host=192.168.10.1

* 验证impala是否正常启动

使用如下命令查看进程是否存在：

ps -ef | grep statestored | grep -v grep; ps -ef | grep catalogd | grep -v grep



使用如下命令查看进程是否存在：

ps -ef | grep impalad| grep -v grep



* 1. **利用impala-shell建表并查询数据**
* 在**rhino001**上建立两个测试样本文件并上传hdfs

[rhino@rhino001 ~]$ vim ~/tab1.csv

|  |
| --- |
| 1,true,123.123,2012-10-24 08:55:00  2,false,1243.5,2012-10-25 13:40:00  3,false,24453.325,2008-08-22 09:33:21.123  4,false,243423.325,2007-05-12 22:32:21.33454  5,true,243.325,1953-04-22 09:11:33s |

[rhino@rhino001 ~]$ vim ~/tab2.csv

|  |
| --- |
| 1,true,12789.123  2,false,1243.5  3,false,24453.325  4,false,2423.3254  5,true,243.325  60,false,243565423.325  70,true,243.325  80,false,243423.325  90,true,243.325 |

[rhino@rhino001 ~]$ hdfs dfs -mkdir -p /user/rhino/sample\_data/tab1

[rhino@rhino001 ~]$ hdfs dfs -mkdir -p /user/rhino/sample\_data/tab2

[rhino@rhino001 ~]$ hdfs dfs -put ~/tab1.csv /user/rhino/sample\_data/tab1/

[rhino@rhino001 ~]$ hdfs dfs -put ~/tab2.csv /user/rhino/sample\_data/tab2/

* 建立两个测试表

[rhino@rhino001 ~]$ ~/impala/sbin/impala-shell.sh

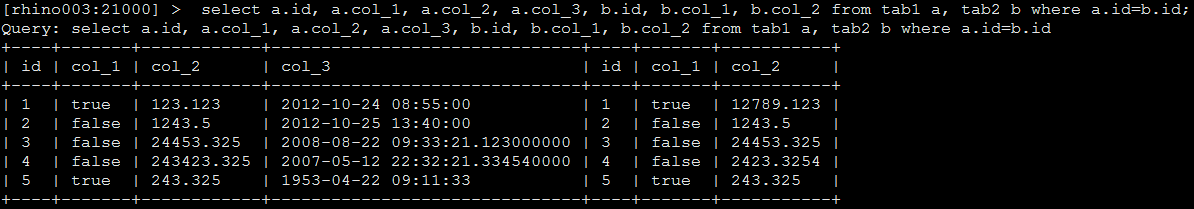
[rhino001:21000]> CREATE EXTERNAL TABLE tab1(id INT, col\_1 BOOLEAN, col\_2 DOUBLE, col\_3 TIMESTAMP) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/user/rhino/sample\_data/tab1';

[rhino001:21000]> CREATE EXTERNAL TABLE tab2(id INT, col\_1 BOOLEAN, col\_2 DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/user/rhino/sample\_data/tab2';

* 查询测试

[rhino001:21000]> select a.id, a.col\_1, a.col\_2, a.col\_3, b.id, b.col\_1, b.col\_2 from tab1 a, tab2 b where a.id=b.id;

查询结果如下图所示：



* 1. **停止impala各进程**
* 在rhino001机器上停止statestored，catalogd进程

[rhino@rhino001 ~]$ ~/impala/sbin/stop-statestored.sh; ~/impala/sbin/stop-catalogd.sh

* 在rhino001机器上停止impalad进程

[rhino@rhino001 ~]$ ~/impala/sbin/stop-impalad.sh

1. **安装spark**
   1. **安装scala**

* 以rhino解压scala安装包(scala-2.11.7.tar.gz)

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf scala-2.11.7.tar.gz

* 配置SCALA\_HOME环境变量

[rhino@rhino001 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export ZOOKEEPER\_HOME=/home/rhino/zookeeper-3.4.5-cdh5.7.0  export HADOOP\_HOME=/home/rhino/hadoop-2.6.0-cdh5.7.0  export HADOOP\_PID\_DIR=/home/rhino/pids  export YARN\_PID\_DIR=/home/rhino/pids  export SCALA\_HOME=/home/rhino/scala-2.11.7  export SENTRY\_HOME=/home/rhino/sentry-1.4.0-cdh5.4.2  export RHINOSQL\_HOME=/home/rhino/rhinosql  export PATH=$JAVA\_HOME/bin:$PATH:$ZOOKEEPER\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$SENTRY\_HOME/bin:$RHINOSQL\_HOME/sbin:$SCALA\_HOME/bin |

* 使环境变量生效

[rhino@rhino001 ~]$ source ~/.bashrc

* 验证scala

[rhino@rhino001 ~]$ scala -version

Scala code runner version 2.11.7 -- Copyright 2002-2013, LAMP/EPFL

* 1. **安装spark**
* 以rhino解压spark安装包(spark-2.2.0-bin-hadoop2.6.tgz)

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001~]$tar -zxvf spark-2.2.0-bin-hadoop2.6.tgz

* 配置SPARK\_HOME环境变量

[rhino@rhino001 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export ZOOKEEPER\_HOME=/home/rhino/zookeeper-3.4.5-cdh5.7.0  export HADOOP\_HOME=/home/rhino/hadoop-2.6.0-cdh5.7.0  export HADOOP\_PID\_DIR=/home/rhino/pids  export YARN\_PID\_DIR=/home/rhino/pids  export SCALA\_HOME=/home/rhino/scala-2.11.7  export SPARK\_HOME=/home/rhino/spark-2.2.0-bin-hadoop2.6  export SPARK\_PID\_DIR=/home/rhino/pids  export SENTRY\_HOME=/home/rhino/sentry-1.4.0-cdh5.4.2  export RHINOSQL\_HOME=/home/rhino/rhinosql  export LD\_LIBRARY\_PATH=/home/rhino/mysql-5.6.25-linux-x86\_64/lib:$LD\_LIBRARY\_PATH:/home/rhino/stclient/lib:/home/rhino/lib  export PATH=$JAVA\_HOME/bin:$PATH:$ZOOKEEPER\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$SENTRY\_HOME/bin:$RHINOSQL\_HOME/sbin:$SCALA\_HOME/bin:$SPARK\_HOME/bin |

* 使环境变量生效

[rhino@rhino001 ~]$ source ~/.bashrc

* 配置slaves节点

[rhino@rhino001 ~]$ cd /home/rhino/spark-2.2.0-bin-hadoop2.6/conf

[rhino@rhino001 conf]$ vim slaves

|  |
| --- |
| rhino001 #注：slave的主机名 |

* 1. **新建并配置hive-site.xml**

[rhino@rhino001 ~]$ cd /home/rhino/spark-2.2.0-bin-hadoop2.6/conf/

[rhino@rhino001 ~]$ vim hive-site.xml

|  |
| --- |
| <configuration>  <property>  <name>hive.metastore.client.socket.timeout</name>  <value>3600</value>  <description>MetaStore Client socket timeout in seconds</description>  </property>  <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:mysql://192.168.10.1:3306/hive\_rhino?createDatabaseIfNotExist=true</value>  </property>  <property>  <name>javax.jdo.option.ConnectionDriverName</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>javax.jdo.option.ConnectionUserName</name>  <value>rhino</value>  </property>  <property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>rhino</value>  </property>  </configuration> |

* 配置spark-en.sh文件

[rhino@rhino001 ~]$ cd /home/rhino/spark-2.2.0-bin-hadoop2.6/conf

[rhino@rhino001 ~]$ cp spark-env.sh.template spark-env.sh

在spark-env.sh文件末添加所需的环境变量配置：

[rhino@rhino001 ~]$ vim spark-env.sh

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export SCALA\_HOME=/home/rhino/scala-2.11.7  export SPARK\_MASTER\_IP=192.168.10.1  export SPARK\_WORKER\_MEMORY=2g  export HADOOP\_CONF\_DIR=/home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop  export SPARK\_CLASSPATH=$SPARK\_CLASSPATH: /home/rhino/spark-2.2.0-bin-hadoop2.6/jars/mysql-connector-java-5.1.35.jar |

* 配置spark-defaults.conf文件

[rhino@rhino001 ~]$ cd /home/rhino/spark-2.2.0-bin-hadoop2.6/conf

[rhino@rhino001 ~]$ cp spark-defaults.conf.template spark-defaults.conf

在spark-defaults.conf文件末添加所需的配置：

[rhino@rhino001 ~]$ vim spark-defaults.conf

|  |
| --- |
| #spark临时目录  spark.local.dir /home/rhino/tmp  spark.sql.warehouse.dir hdfs://rhino001:9020/user/hive/warehouse |

* **注：YARN方式不需要启动spark。即以下命令不需要操作。**
* 启动spark并查看信息

[rhino@rhino001 ~]$ cd ~/spark-2.2.0-bin-hadoop2.6/sbin

[rhino@rhino001 ~] ./start-all.sh

[rhino@rhino001 ~]jps

* 停止spark分布式集群

[rhino@rhino001 ~] ./stop-all.sh

* 1. **上传jar包（重要）**
* 将spark环境中的jar包放到HDFS

[rhino@rhino001 ~]$hdfs dfs -mkdir hdfs://rhino001:9020/sharkjars-2.0/

[rhino@rhino001 ~]$hdfs dfs -put ~/spark-2.2.0-bin-hadoop2.6/jars/\* hdfs://rhino001:9020/sharkjars-2.0/

* 1. **修改yarn文件**
* 修改yarn文件（如果缺少的内容已存在，则无需修改）

[rhino@rhino001 ~]$vim ~/hadoop-2.6.0-cdh5.7.0/bin/yarn

|  |
| --- |
| elif [ "$COMMAND" = "nodemanager" ] ; then  CLASSPATH=${CLASSPATH}:$YARN\_CONF\_DIR/nm-config/log4j.properties  **for f in ${SPARK\_HOME}/yarn/\*.jar; do**  **CLASSPATH=${CLASSPATH}:${f}**  **done**  **#在下面这句前面增加上述内容**  CLASS='org.apache.hadoop.yarn.server.nodemanager.NodeManager' |

1. **部署YARN**
   * 1. **配置yarn**

* 配置yarn-site.xml

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/

[rhino@rhino001 hadoop]$ vim yarn-site.xml

**全文复制以下内容，并做相应的修改.**

|  |
| --- |
| <?xml version="1.0"?>  <configuration>  <property>  <name>yarn.resourcemanager.address</name>  <value>rhino001:8032</value>  </property>  <property>  <name>yarn.resourcemanager.scheduler.address</name>  <value>rhino001:8030</value>  </property>  <property>  <name>yarn.resourcemanager.resource-tracker.address</name>  <value>rhino001:8031</value>  </property>  <property>  <name>yarn.resourcemanager.webapp.address</name>  <value>rhino001:8088</value>  </property>  <property>  <name>yarn.resourcemanager.webapp.https.address</name>  <value>rhino001:8090</value>  </property>  <property>  <name>yarn.resourcemanager.scheduler.class</name>  <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>  </property>  <property>  <name>yarn.scheduler.fair.allocation.file</name>  <value>/home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/fair-scheduler.xml</value>  </property>  <property>  <name>yarn.scheduler.fair.preemption</name>  <value>true</value>  </property>  <property>  <name>yarn.scheduler.fair.assignmultiple</name>  <value>true</value>  </property>  <property>  <name>yarn.scheduler.fair.max.assign</name>  <value>1</value>  </property>  <property>  <name>yarn.scheduler.fair.user-as-default-queue</name>  <value>false</value>  <description>default is True</description>  </property>  <property>  <name>yarn.scheduler.fair.allow-undeclared-pools</name>  <value>false</value>  <description>default is True</description>  </property>  <property>  <name>yarn.acl.enable</name>  <value>true</value>  </property>  <property>  <name>yarn.admin.acl</name>  <value>rhino</value>  </property>  <property>  <name>yarn.scheduler.minimum-allocation-mb</name>  <value>20480</value>  <discription>单个任务可申请最少内存，默认1024MB</discription>  </property>  <property>  <name>yarn.scheduler.maximum-allocation-mb</name>  <value>64000</value>  <discription>单个任务可申请最大内存，默认8192MB</discription>  </property>  <property>  <name>yarn.nodemanager.resource.memory-mb</name>  <value>81920</value>  <discription>单个任务可申请最大内存，默认8192MB</discription>  </property>  <property>  <name>yarn.scheduler.maximum-allocation-vcores</name>  <value>20</value>  </property>  <property>  <name>yarn.nodemanager.resource.cpu-vcores</name>  <value>20</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>spark\_shuffle,mapreduce\_shuffle</value>  <description>开启动态分配时需要</description>  </property>  <property>  <name>yarn.nodemanager.aux-services.spark\_shuffle.class</name>  <value>org.apache.spark.network.yarn.YarnShuffleService</value>  <description>开启动态分配时需要</description>  </property>  <property>  <name>yarn.nodemanager.aux-services.mapreduce\_shuffle.class</name>  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  </property>  </configuration> |

* 新建并配置fair-scheduler.xml

[rhino@rhino001 ~]$ cd /home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/

[rhino@rhino001hadoop]$ vim fair-scheduler.xml

**(root是总资源，是其余队列资源之和。注意文件格式不能被破坏！)**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <allocations>  <queue name="root">  <minResources>1000 mb,2 vcores</minResources>  <maxResources>81920 mb,40 vcores</maxResources>  <maxRunningApps>10</maxRunningApps>  <schedulingMode>fair</schedulingMode>  <aclSubmitApps> </aclSubmitApps>  <aclAdministerApps> </aclAdministerApps>  <queue name="spark-max">  <minResources>1000 mb,4 vcores</minResources>  <maxResources>61440 mb,30 vcores</maxResources>  <maxRunningApps>10</maxRunningApps>  <aclSubmitApps>rhino,admin</aclSubmitApps>  <weight>2.0</weight>  <schedulingPolicy>fair</schedulingPolicy>  </queue>  <queue name="spark-mid">  <minResources>1000 mb,4 vcores</minResources>  <maxResources>20480 mb,10 vcores</maxResources>  <maxRunningApps>10</maxRunningApps>  <aclSubmitApps>rhino,admin</aclSubmitApps>  <weight>1.0</weight>  <schedulingPolicy>fair</schedulingPolicy>  </queue>  </queue>  </allocations> |

**说明：**

1. fair-scheduler.xml配置是yarn资源分配采用fair模式下配置文件；
2. 请根据实际情况配置队列个数，以及队列中的配置值；
3. aclSubmitApps和aclAdministerApps标签中如果没有用户，请添加空格。
   * 1. **启动YARN**

* 以rhino用户启动yarn

[rhino@rhino001 ~]$ start-yarn.sh

* 检查YARN是否正常启动

使用浏览器登录到yarn监控页面 192.168.10.1:8088，查看Active Nodes是否为1(即配置的slaves节点个数)，如果为1即说明yarn已正常启动。

* 停止YARN(需要停止时执行)

[rhino@rhino001 ~]$ stop-yarn.sh

1. **部署Kafka**
   1. **安装Kafka**

* 以rhino用户在rhino001机器下解压安装包kafka\_2.11-1.1.0.tgz

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf kafka\_2.11-1.1.0.tgz

注：如果安装oceansource，kafka版本至少kafka\_2.11-1.1.0。

* 1. **配置kafka**
* 配置server.properties

[rhino@rhino001 ~]$ cd /home/rhino/kafka\_2.11-1.1.0

[rhino@rhino001 kafka\_2.11-0.10.1.0]$ vim config/server.properties

|  |
| --- |
| # The id of the broker. This must be set to a unique integer for each broker.  broker.id=0  # Hostname and port the broker will advertise to producers and consumers. If not set,  # it uses the value for "listeners" if configured. Otherwise, it will use the value  # returned from java.net.InetAddress.getCanonicalHostName().  advertised.listeners=PLAINTEXT://rhino001:9092  # A comma seperated list of directories under which to store log files  log.dirs=/data01/kafka-logs,/data02/kafka-logs,/data03/kafka-logs  （如果有多块磁盘，并对kafka的性能有较高要求，建议在每个磁盘下都配置一个目录）  # Zookeeper connection string (see zookeeper docs for details).  zookeeper.connect=rhino001:2181 |

* 1. **启动kafka**
* 启动kafka服务

[rhino@rhino001 ~]$ cd /home/rhino/kafka\_2.11-1.1.0

[rhino@rhino001 kafka\_2.11-0.10.1.0]$ bin/kafka-server-start.sh -daemon config/server.properties

* 验证进程是否正常启动

使用 “ps -ef | grep kafka| grep -v grep” 命令观察进程是否存在。

* 验证消息发送和接收

**发送消息：**

[rhino@rhino001 ~]$ cd kafka\_2.11-1.1.0

[rhino@rhino001 kafka\_2.11-1.1.0]$ bin/kafka-console-producer.sh --broker-list rhino104:9092 --topic test

在控制台输入一些消息，按ctrl+c退出发送。

**读取消息：**

[rhino@rhino001~]$ cd kafka\_2.11-1.1.0

[rhino@rhino001 kafka\_2.11-1.1.0]$ bin/kafka-console-consumer.sh --bootstrap-server rhino104:9092 --topic test --from-beginning

控制台中看到之前发送的消息则证明kafka配置成功，按ctrl+c退出控制台。

* 1. **停止kafka**
* 停止kafka进程（需要停止时执行）

[rhino@rhino001~]$ cd kafka\_2.11-1.1.0

[rhino@rhino001 kafka\_2.11-1.1.0]$bin/kafka-server-stop.sh

1. **安装Redis**

当前后台通讯选择redis，或使用**流式告警**且配置告警阈值，需要安装Redis。

* 1. **安装**

|  |
| --- |
| * 以rhino 用户解压redis安装文件   [rhino@rhino001 ~]$cd ~  [rhino@rhino001 ~]tar -zxvf redis-4.0.10.tar.gz   * make编译程序:   [rhino @rhino061 ~]$ cd redis-4.0.10/  [rhino @rhino061 redis-4.0.10]$ make   * make安装程序:   切换到root用户下执行make install   * 新建redis目录   rhino用户下执行：  mkdir -p /home/rhino/redis/bin  mkdir -p /home/rhino/redis/conf  root用户下执行：  mv /usr/local/bin/\* /home/rhino/redis/bin  chown rhino:rhino /home/rhino/redis/bin/\*  rhino用户下执行：  mv /home/rhino/redis-4.0.10/redis.conf /home/rhino/redis/conf |

* 1. **配置**

1. vim /home/rhino/redis/conf/redis.conf

|  |
| --- |
| # By default Redis does not run as a daemon. Use 'yes' if you need it.  # Note that Redis will write a pid file in /var/run/redis.pid when daemonized.  daemonize yes  # IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES  # JUST COMMENT THE FOLLOWING LINE.  # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  bind 162.168.1.10  # requirepass foobared，指定密码，也可以不指定，如果指定，登录客户端时候需要密码认证  requirepass 123456 |

2. 配置环境变量

vim ~/.bashrc,加上如下内容：

|  |
| --- |
| export REDIS\_HOME=/home/rhino/redis  export PATH里加上$REDIS\_HOME/bin |

保存环境变量：

source ~/.bashrc

* 1. **启动和停止**
* 启动redis服务端

|  |
| --- |
| [rhino@rhino061 ~]$ redis-server /home/rhino/redis/conf/redis.conf |

然后通过ps -ef|grep redis命令或者安装目录下的logs目录查看进程是否启动成功。

* 启动redis客户端

|  |
| --- |
| 输入redis-cli命令启动客户端  [rhino@rhino061 ~] $ redis-cli -h 162.168.1.10  162.168.1.10:6379> auth 123456 #密码认证，如果没有密码，此步可以省略  OK  162.168.2.61:6379> info  #启动后输入info查看redis相关信息 |

* 停止redis服务

|  |
| --- |
| [rhino@rhino061 ~] $ redis-cli shutdown |

1. **安装ftp服务器**
   1. **安装**

* 将CentOS-7-x86\_64-Everything-1611.iso在/root/目录下，挂载镜像文件：

|  |
| --- |
| mkdir /mnt/cdrom/  mount -o loop /root/CentOS-7-x86\_64-Everything-1611.iso /mnt/cdrom/ |

其中，系统镜像文件版本一定要和系统版本匹配，不然后续安装会报错。

* 安装sftp和ftp

[rhino@rhino001 ~]$ cd /mnt/cdrom/Packages

[rhino@rhino001 ~]$ rpm -ivh vsftpd-3.0.2-10.el7.x86\_64 rpm

[rhino@rhino001 ~]$ rpm -ivh ftp-0.17-66.el7.x86\_64.rpm

rpm文件根据具体操作系统版本会有不同。

* 1. **配置连接数**
* 以root用户登录，配置ftp最大连接数

[root@rhino001 ~]# vim /etc/vsftpd/vsftpd.conf

|  |
| --- |
| max\_clients=300#指明服务器总的客户并发连接数为300  max\_per\_ip=50#指明每个客户机的最大连接数为50 |

* 配置sftp最大连接数

[root@rhino001 ~]# vim/etc/ssh/sshd\_config

|  |
| --- |
| #MaxStartups 10:30:100  MaxStartups 100 |

* 重启sshd服务

[root@rhino001 ~]]#service sshd restart

* 1. **启动ftp服务**
* 启动服务

[root@rhino001 ~]]#service vsftpd restart

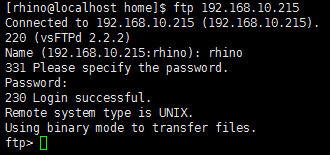
* 从其他ftp服务器连接到该服务器验证是否正常（假设rhino002机器下已经安装ftp服务器）

分别以ftp和sftp建立连接：

[rhino@rhino002 ~]$ftp rhino001 （验证ftp）

[rhino@rhino002 ~]$ sftp rhino001 （验证sftp）

输入用户名rhino，密码rhino，出现以下信息即为成功，输入quit、exit或bye断开连接。



1. **部署ElasticSearch6.1.3**
   1. **安装ElasticSearch6.1.3**

以rhino用户解压ElasticSearch安装文件：Elasticsearch-6.1.3.tar.gz解压到/home/rhino/目录下。

[rhino@rhino001 ~]$ cd /home/rhino

[rhino@rhino001 ~]$ tar -zxvf elasticsearch-6.1.3.tar.gz

* 1. **修改配置**

ES服务使用9200、9300端口提供服务，安装前需要保证端口未被占用。

静态配置

根据实际服务器配置情况设置ES服务可以使用的内存，ES要求至少2g，实际生产环境应该配置30g。

[rhino@rhino001 ~]$vi /home/rhino/elasticsearch-6.1.3/config/jvm.options

|  |
| --- |
| -Xms30G  -Xmx30G |

[rhino@rhino001 ~]$vi /home/rhino/elasticsearch-6.1.3/config/elasticsearch.yml

|  |
| --- |
| #锁内存配置  bootstrap.memory\_lock: true  #配置服务器ip  network.host：192.168.10.1  #集群初始化节点列表  discovery.zen.ping.unicast.hosts: ["rhino001"]  #elasticsearch数据存储路径，默认为安装目录下的data目录  path.data: /data01/esdata,/data02/esdata,/data03/esdata  #启用基于磁盘的分片分配，true为启用，false为禁止  cluster.routing.allocation.disk.threshold\_enabled: true  #节点中磁盘空间超过设定值，节点不再分配新的分片  cluster.routing.allocation.disk.watermark.low: 95%  #节点中磁盘空间超过设定值，节点开始尝试重新分配分片  cluster.routing.allocation.disk.watermark.high: 95%  #更新节点上磁盘使用信息的时间间隔  cluster.info.update.interval: 30s |

注：修改上面这个文件时，每个分号（:）后面需要有个空格。

**动态配置（创建索引时使用，安装启动时无需关注）**

动态配置要在索引建好之后动态修改，命令如下：

|  |
| --- |
| curl -XPUT 'localhost:9200/index1/\_settings' -d '{"index.refresh\_interval":"10s"}' |

可以修改的配置项如下：

|  |
| --- |
| #索引刷新时间  index.refresh\_interval: 10s  #修改索引备份数，1表示集群中有2份数据  index.number\_of\_replicas: 1 |

* 1. **启动**

cd /home/rhino/elasticsearch-6.1.3/bin

./elasticsearch -d

然后通过ps -ef|grep elasticsearch命令或者安装目录下的logs目录查看进程是否启动成功。

* 1. **常用REST接口**

在shell命令行下，通过如下命令查看集群和索引

查看整个集群状态：

|  |
| --- |
| curl -XGET 'http://localhost:9200/\_cluster/stats?human&pretty' |

查看集群线程数，相当于集群负载：

|  |
| --- |
| curl -XGET http://localhost:9200/\_cat/thread\_pool?v |

查看索引分片数：

|  |
| --- |
| curl -XGET http://localhost:9200/\_cat/shards/index1?v |

查看文档结构：

|  |
| --- |
| curl -XGET 'localhost:9200/index1/\_mapping/type1' |

查看索引段：

|  |
| --- |
| curl -XGET 'http://localhost:9200/index1/\_segments?pretty' |

合并索引段：

|  |
| --- |
| curl -XPOST 'http://localhost:9200/index1/\_forcemerge?max\_num\_segments=5' |

* 1. **停止**

通过ps -ef|grep elasticsearch命令查看进程号，通过kill -9 进程id，直接杀掉elasticsearch进程

1. **HBASE安装**
   1. **安装hbase**

以rhino用户在rhino058机器下解压hbase安装包（本次安装为集群环境安装hbase单机模式，单机环境安装需要对hbase.rootdir、hbase.zookeeper.quorum配置项修改实际值）

[rhino@rhino058 ~]$ tar -zxvf hbase-1.4.9-bin.tar.gz

* 1. **配置hbase**

修改hbase的conf下的hbase-site.xml文件

|  |
| --- |
| <?xml version="1.0"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>hbase.rootdir</name>  <value>hdfs://sinovatiocluster/hbase</value>  </property>  <property>  <name>hbase.cluster.distributed</name>  <value>true</value>  </property>  <property>  <name>hbase.zookeeper.property.dataDir</name>  <value>/home/rhino/hbasezookeeper</value>  </property>  <property >  <name>hbase.fs.tmp.dir</name>  <value>/user/hbase-staging</value>  </property>  <property >  <name>hbase.bulkload.staging.dir</name>  <value>${hbase.fs.tmp.dir}</value>  </property>  <property>  <name>fs.hdfs.impl</name>  <value>org.apache.hadoop.hdfs.DistributedFileSystem</value>  </property>  <property>  <name>hbase.zookeeper.quorum</name>  <value>162.168.3.56:2181,162.168.3.57:2181,162.168.3.58:2181</value>  </property>  <property>  <name>hbase.regionserver.handler.count</name>  <value>10</value>  </property>  <!--<property>  <name>hbase.client.write.buffer</name>  <value>5242880</value>  </property>-->  <!--region size 10g -->  <property>  <name>hbase.hregion.max.filesize</name>  </property>  <property>  <name>hbase.hstore.blockingStoreFiles</name>  <value>30</value>  </property>  <!--内存能缓存的数据最大字节数，当memstore中缓存的数据达到这个值，hbase就将数据flush到硬盘上 默认128m-->  <property>  <name>hbase.hregion.memstore.flush.size</name>  <value>134217728</value>  <!-- <value>33554432</value>-->  </property>  </configuration> |

修改hbase的hbase-env.sh文件

/home/rhino/ hbase-1.4.9/conf/ hbase-env.sh文件添加如下配置配置：

|  |
| --- |
| export JAVA\_HOME=/home/rhino/jdk1.8.0\_181  export HBASE\_HEAPSIZE=10G  export HBASE\_OPTS="-XX:+UseConcMarkSweepGC"  export HBASE\_MASTER\_OPTS="$HBASE\_MASTER\_OPTS -XX:PermSize=512m -XX:MaxPermSize=512m"  export HBASE\_REGIONSERVER\_OPTS="$HBASE\_REGIONSERVER\_OPTS -Xms30720m -Xmx30720m -XX:PermSize=512m -XX:MaxPermSize=512m"  export HBASE\_MANAGES\_ZK=false |

注：export HBASE\_MANAGES\_ZK=false(表示使用自己的zookeeper)

修改regionservers文件

[rhino@rhino058 conf]$ vi /home/rhino/hbase-1.4.9/conf/regionservers

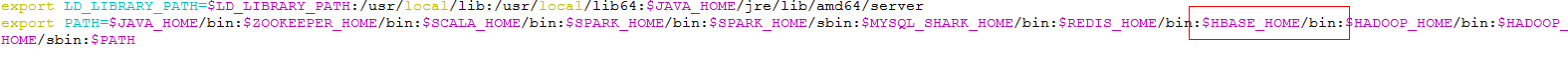
|  |
| --- |
| rhino058 |

配置环境变量：

[rhino@rhino058 ~]$ vi .bashrc

export HBASE\_HOME=/home/rhino/hbase-1.4.9

PATH中添加:$HBASE\_HOME/bin



source ~/.bashrc

* 1. **启动 hbase**

1. cd $HBASE\_HOME/bin

启动hbase

./start-hbase.sh

关闭hbase

./stop-hbase.sh

* 1. **验证hbase**

|  |
| --- |
| $HBASE\_HOME/bin/hbase shell |
| 查看hbase服务器状态 |
| hbase(main):001:0> status  3 servers, 0 dead, 0.6667 average load |
| 创建一个表 |
| hbase(main):001:0>create "t","t\_id","t\_vl"  0 row(s) in 2.4560 seconds |
| 插入数据 |
| hbase(main):002:0>put "t","10001","t\_vl:name","jss"  0 row(s) in 0.2620 seconds  hbase(main):003:0>put "t","10001","t\_vl:age","99"  0 row(s) in 0.0330 seconds  hbase(main):004:0>put "t","10001","t\_vl:gengeral1","man"  0 row(s) in 0.0350 seconds |
| 查询数据 |
| hbase(main):005:0> get "t","10001"  COLUMN CELL  t\_vl:age timestamp=1464751483159, value=99  t\_vl:gengeral1 timestamp=1464751514376, value=man  t\_vl:name timestamp=1464751462542, value=jss  3 row(s) in 0.0890 seconds  hbase(main):006:0>get "t","10001","t\_vl:age"  COLUMN CELL  t\_vl:age timestamp=1464751483159, value=99  1 row(s) in 0.0260 seconds |
| 全表扫描 |
| hbase(main):007:0> scan "t"  ROW COLUMN+CELL  10001 column=t\_vl:age, timestamp=1464751483159, value=99  10001 column=t\_vl:gengeral1, timestamp=1464751514376, value=man  10001 column=t\_vl:name, timestamp=1464751462542, value=jss  1 row(s) in 0.0640 seconds |

1. **hive库安装**
   1. **解压安装**

解压安装包：tar -zxvf apache-hive-2.3.2-bin.tar.gz至~

* 1. **修改配置**

配置环境变量：vi ~/.bashrc 添加以下两行内容：

   export HIVE\_HOME=/home/rhino/apache-hive-2.3.3-bin

   export PATH=$PATH:$HIVE\_HOME/bin

保存退出。source ~/.bashrc

进入解压好的apache-hive-2.3.2-bin目录找到conf目录，将hive-default.xml.template文件拷贝一份，并且重命名为hive-site.xml，将一下配置加入到hive-site.xml 文件中开头 （标签<value>中的内容自己定义）

|  |
| --- |
| <property>  <name>system:java.io.tmpdir</name>  <value>/home/rhino/apache-hive-2.3.3-bin/tmpdir</value>  </property>  <property>  <name>system:user.name</name>  <value>hive</value>  </property> |

在mysql中创建hive库，用于存放元数据：create database hive;

获取mysql的jdbc驱动包放在lib目录（/home/rhino/apache-hive-2.3.3-bin/lib）下：

/home/rhino/spark-2.2.0-bin-hadoop2.6/jars下有ll mysql-connector-java-5.1.35.jar

再次进入到hive目录中的conf文件中，配置hive-site.xml文件

|  |
| --- |
| 分别找到  （1）javax.jdo.option.ConnectionURL， 553  （2）javax.jdo.option.ConnectionDriverName， 1028  （3）javax.jdo.option.ConnectionUserName， 1053  （4）javax.jdo.option.ConnectionPassword 538  这四项配置，其中这四项的<value>分别填：  （1）<value>jdbc:mysql://162.168.2.61:3306/hive?characterEncoding=utf8&amp;useSSL=false</value>，  （2）<value>com.mysql.jdbc.Driver</value>，  （3）<value>rhino</value>，  （4）<value>rhino</value> |

初始化元数据库-hive库生成相关的表：

|  |
| --- |
| 进入hive目录的bin文件中使用  ./schematool -dbType mysql -initSchema 进行元数据库初始化 |

* 1. **启动hive**

初始化完成后，使用./hive命令启动hive，出现hive>的时候，就可以使用了。

启动mestore

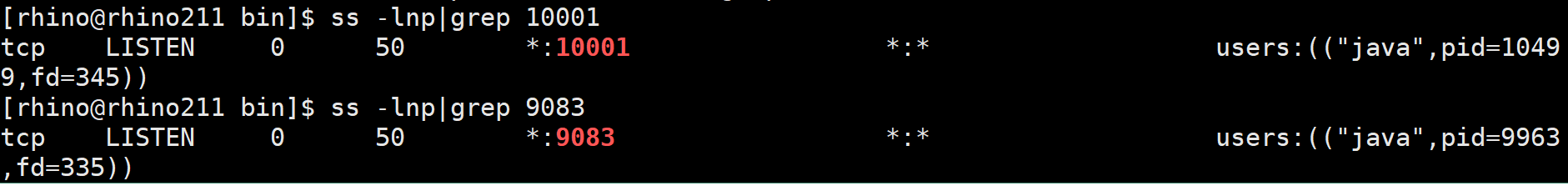
[rhino@rhino151 bin]$ nohup ./hive --service metastore &

作用：Starting Hive Metastore Server

启动hive

[rhino@rhino151bin]$ nohup ./hive --service hiveserver2 --hiveconf hive.server2.thrift.port=10001 &

作用：Starting HiveServer2



|  |
| --- |
| 注意：/home/rhino/hadoop-2.6.0-cdh5.7.0/etc/hadoop/core-site.xml中添加  <property>  <name>hadoop.proxyuser.rhino.hosts</name>  <value>\*</value>  </property>  <property>  <name>hadoop.proxyuser.rhino.groups</name>  <value>\*</value>  <description>  The 'nfsserver' user is allowed to proxy all members of the 'users-group1' and  'users-group2' groups. Note that in most cases you will need to include the  group "root" because the user "root" (which usually belonges to "root" group) will  generally be the user that initially executes the mount on the NFS client system.  Set this to '\*' to allow nfsserver user to proxy any group.  </description>  </property>  <property>  <name>hadoop.proxyuser.rhino.hosts</name>  <value>\*</value>  <description>  This is the host where the nfs gateway is running. Set this to '\*' to all requests from any hosts to be proxied.  </description>  </property> |

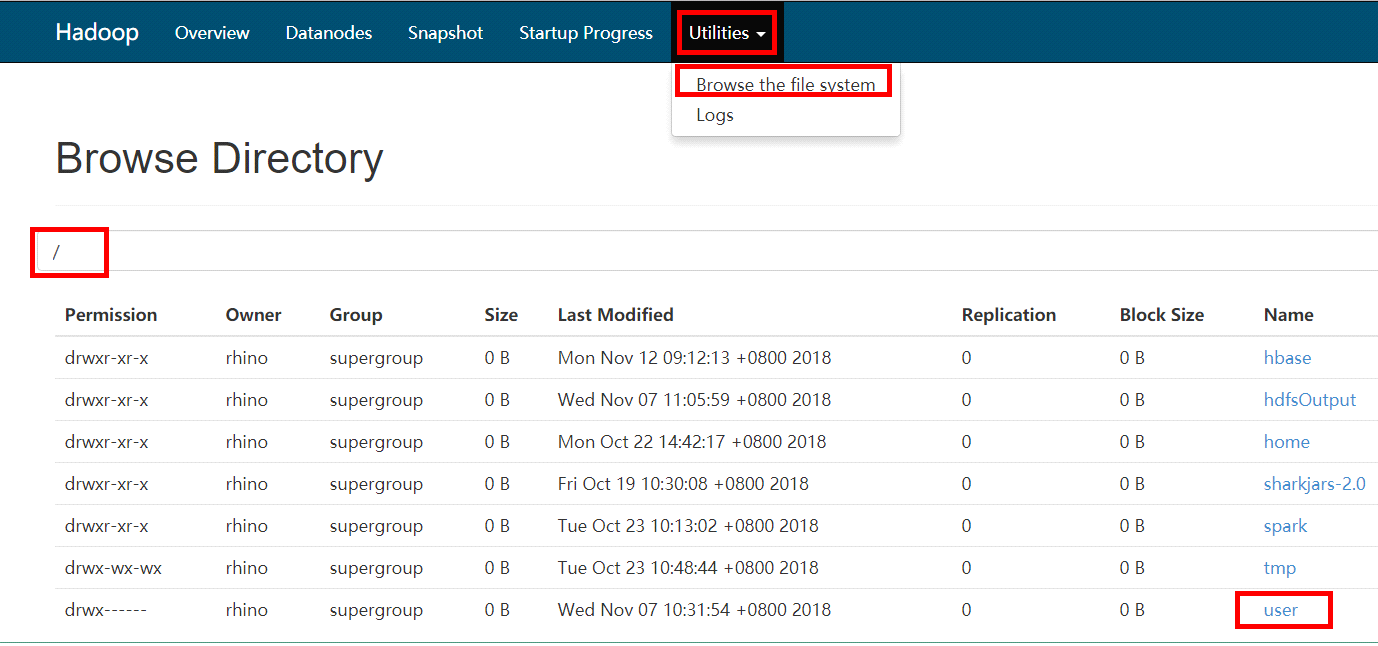
修改保存后重启hdfs：stop-dfs.sh ;start-dfs.sh

* 1. **验证hive**

验证登陆hive库执行：beeline -u jdbc:hive2://162.168.2.61:10001/

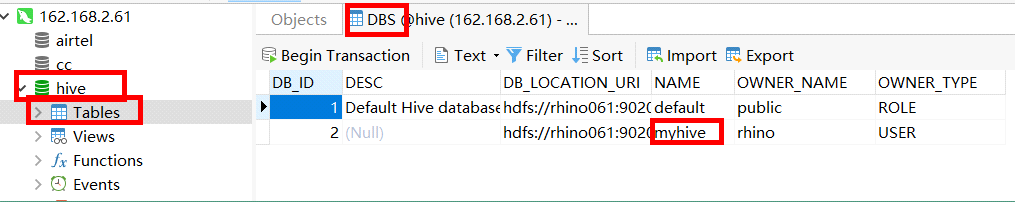
创建一个数据库可以在页面查看:

http://162.168.2.61:50070

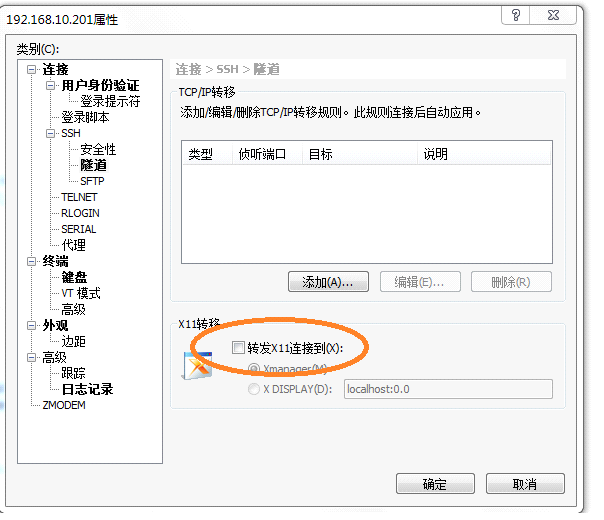




同时在mysql也增加了一条记录：



1. **FAQ**
2. 点击“运行”按钮时，出现”Xmanager运行失败;请检查此系统的Xmanager是否已正确安装”的提示框；

解决方法：利用XShell登录部署环境时,需要做如下操作：如下图所示，将框中的选项勾掉（不要打勾）

1. **附录一. 安装本地yum源**

[root@ rhino001 ~]# mkdir /mnt/cdrom/ ------》创建光盘挂载目录

假设/dev/cdrom为光盘所在目录

[root@ rhino001 ~]# mount /dev/cdrom /mnt/cdrom/ --------》挂载光盘

若挂载root目录下的镜像文件，则执行

mount -o loop /root/CentOS-6.5-x86\_64-bin-DVD1.iso /mnt/cdrom/

[root@ rhino001 ~]# cd /etc/yum.repos.d/ --------》切换到YUM配置目录

[root@ rhino001 yum.repos.d]# tar czf repo.tar.gz ./\* --------》将原来所有的文件进行打包备份

[root@ rhino001 yum.repos.d]# rm -f CentOS-Base.repo CentOS-Debuginfo.repo CentOS-Vault.repo --------》删除文件，只保留CentOS-Media.repo

[root@ rhino001 yum.repos.d]# vim CentOS-Media.repo

|  |
| --- |
| [c6-media]  name=CentOS-$releasever - Media  baseurl=file:///mnt/cdrom/ ##这里为本地源路径  gpgcheck=1  enabled=1##开启本地源  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6 |

[root@ rhino001 yum.repos.d]#yum clean all --------》清除原先yum缓存