帷策公司Hadoop+Spark集群

目录

**[一、 什么是hadoop](#_Toc28252_WPSOffice_Level1)** **[1](#_Toc28252_WPSOffice_Level1)**

**[二、Spark与hadoop关系](#_Toc30461_WPSOffice_Level1)** **[2](#_Toc30461_WPSOffice_Level1)**

**[三、公司hadoop配置详情](#_Toc8880_WPSOffice_Level1)** **[2](#_Toc8880_WPSOffice_Level1)**

[Hadoop服务器列表](#_Toc30461_WPSOffice_Level2) [2](#_Toc30461_WPSOffice_Level2)

**[四、部署步骤](#_Toc11560_WPSOffice_Level1)** **[3](#_Toc11560_WPSOffice_Level1)**

**[步骤一、环境准备](#_Toc6841_WPSOffice_Level1)** **[3](#_Toc6841_WPSOffice_Level1)**

**[步骤二、在Namenode01主机中修改hadoop配置文件，然后复制到其它各主机](#_Toc18949_WPSOffice_Level1)** **[4](#_Toc18949_WPSOffice_Level1)**

**[步骤三、在namenode01中启动HDFS和YARN](#_Toc14164_WPSOffice_Level1)** **[9](#_Toc14164_WPSOffice_Level1)**

[1）、在namenode01中启动HDFS](#_Toc8880_WPSOffice_Level2) [9](#_Toc8880_WPSOffice_Level2)

[2）、启动Yarn](#_Toc11560_WPSOffice_Level2) [10](#_Toc11560_WPSOffice_Level2)

**[步骤四、配置spark](#_Toc5184_WPSOffice_Level1)** **[12](#_Toc5184_WPSOffice_Level1)**

**[步骤五、集群操作](#_Toc2982_WPSOffice_Level1)** **[15](#_Toc2982_WPSOffice_Level1)**

[1） 、服务管理：](#_Toc6841_WPSOffice_Level2) [15](#_Toc6841_WPSOffice_Level2)

[2）、集群文件管理及常用命令使用：](#_Toc18949_WPSOffice_Level2) [16](#_Toc18949_WPSOffice_Level2)

**[步骤六、节点管理](#_Toc17637_WPSOffice_Level1)** **[18](#_Toc17637_WPSOffice_Level1)**

**[一、新增数据节点](#_Toc7425_WPSOffice_Level1)** **[18](#_Toc7425_WPSOffice_Level1)**

[A：在新增节点修改主机名datanode02，安装jdk环境](#_Toc14164_WPSOffice_Level2) [18](#_Toc14164_WPSOffice_Level2)

[B：Namenode01节点中设置到新节点的无密码登陆](#_Toc5184_WPSOffice_Level2) [18](#_Toc5184_WPSOffice_Level2)

[C：在Namenode01节点slaves文件中添加新节点](#_Toc2982_WPSOffice_Level2) [18](#_Toc2982_WPSOffice_Level2)

[D：在所有节点/etc/hosts文件中增加新节点（所有节点保持一致）](#_Toc17637_WPSOffice_Level2) [18](#_Toc17637_WPSOffice_Level2)

[E：在新节点中执行](#_Toc7425_WPSOffice_Level2) [18](#_Toc7425_WPSOffice_Level2)

**[二、删除数据节点](#_Toc8854_WPSOffice_Level1)** **[21](#_Toc8854_WPSOffice_Level1)**

[步骤](#_Toc8854_WPSOffice_Level2)

**[二、 更改hadoop管理服务器的pid文件路径](#_Toc23500_WPSOffice_Level1)** **[22](#_Toc23500_WPSOffice_Level1)**

该文档简单介绍了hadoop组成，并用于构建和维护公司的Hadoop+Spark集群

1. **什么是hadoop**

Hadoop是[Apache软件基金会](https://www.apache.org/" \t "https://www.cnblogs.com/wing1995/p/_blank)支持可靠的、可扩展的一个开源的分布式计算框架的工程。

Apache Hadoop软件库是一个允许使用简单编程模型跨计算机集群处理大型数据集合的框架，其设计**的初衷是将单个服务**器扩展成上千个机器组成的一个集群为大数据提供计算服务，其中每个机器都提供本地计算和存储服务

Hadoop由多个组件构成，核心组件如下：

**Hadoop Common**：支持其他Hadoop模块的通用工具。

**Hadoop Distributed File System(HDFS)**：提供高吞吐量的访问应用数据的一个分布式文件系统。

**Hadoop YARN**：一种作业调度和集群资源管理的框架。

**Hadoop MapReduce**：一种基于Yarn来处理大数据集合的系统。

**[Spark](http://spark.apache.org/" \t "https://www.cnblogs.com/wing1995/p/_blank)**：一种用于Hadoop数据的快速通用计算引擎。Spark提供一种支持广泛应用的简单而易懂的编程模型，包括ETL（ Extract-Transform-Load）、机器学习、流处理以及图计算。

**HIVE**：数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供类SQL查询功能，操作接口采用类SQL语法，提供快速开发的能力，避免了去写MapReduce，减少开发人员的学习成本。Hive目前底层还是用MapReduce，以后可能会用Spark,Tez。Hive差不多是一个查询接口，你的SQL语句还是会翻译成MapReduce任务的，你用explain就可以看到这些任务的执行计划。下面介绍下hive的基础安装过程以及简单的使用。

除此还有多个组件**Avro、Hive、Hbase、Pig、Zookeeper、Mahout、Casscandra、Tez等,最少需要3台服务器才能搭建hadoop集群**

**二、Spark与hadoop关系**

如果说MapReduce是第一代计算引擎，那么Spark就是第二代计算引擎。Spark优点：速度快、易用性、通用性、支持多种模式运行

三、**公司hadoop配置详情**

公司Hadoop版本hadoop-2.7.7,spark版本spark-2.3.2

**Hadoop服务器列表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 服务器IP | 主机名 | 任务列表 | 启用端口 | 安装目录 | 软件 |
| 192.168.0.201 | Namenode01 | NameNode  SecondaryNamenode  ResourceManager | 50070  50090  8088  9000 | /opt/hadoop-2.7.7  /opt/spark-2.3.2  /opt/apache-hive-3.1.2-bin | HDFS  Yarn  Hive |
| 192.168.0.203 | Datanode03 | DataNode  NodeManager | 50075  8042 | /opt/hadoop-2.7.7 | HDFS |
| 192.168.0.204 | Datanode03 | DataNode  NodeManager | 50075  8042 | /opt/hadoop-2.7.7 | HDFS |
| 192.168.0.205 | Datanode03 | DataNode  NodeManager | 50075  8042 | /opt/hadoop-2.7.7 | HDFS |
| 192.168.0.206 | Datanode04 | DataNode  NodeManager | 50075  8042 | /opt/hadoop-2.7.7 | HDFS |
| 192.168.0.207 | Datanode | DataNode  NodeManager | 50075  8042 | /opt/hadoop-2.7.7 | HDFS |

**四、部署步骤**

**步骤一、环境准备**

**1）**hadoop是用java编写的，spark是由Scala编写的，所以还需要安装java-1.8.0-openjdk-devel和scala，hadoop需要集群所有设备安装，即集群所有设备需要安装jdk环境，spark只需要Namenode01安装

yum -y install java-1.8.0-openjdk-devel #安装jdk

java -version #查看版本

rpm -ql java-1.8.0-openjdk-devel #查看安装路径

剩余软件也可手动下载，但要注意各版本之间能否兼容，以下为网址：

**JDK**: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

**Hadoop**：http://archive.apache.org/dist/hadoop/common/

**Scala**：<https://www.scala-lang.org/download/all.html>

**Spark**：<http://archive.apache.org/dist/spark/spark-2.2.0/>

http://spark.apache.org/

**2）**Namenode01与其它主机之间免密登录，在Namenode01运行：

ssh-keygen -f /root/.ssh/id\_rsa -N ‘ ’ #生成秘钥文件

ssh-copy-id 192.168.0.205/206/207 #传秘钥文件到各主机

**\*注：防火墙需要关闭或者设置成信任区：**

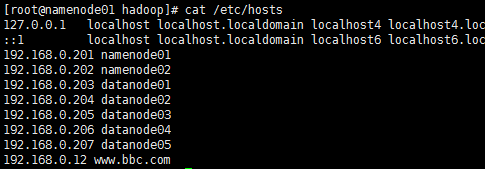
**firewall-cmd --set-default-zone=trusted**

**Setenforce=0 和修改文件**

**vim /etc/selinux/config SELINUX=disabled**

3）做好各主机的域名解析，每台主机运行：

vim /etc/hosts



\*正式环境还需做NTP时间同步以namenode01为时间服务器，vim /etc/chrony.conf

**步骤二、在Namenode01主机中修改hadoop配置文件，然后复制到其它各主机**

**1）、将下载好的压缩包解压至/opt/hadoop-2.7.7**

tar -xvf /root/hadoop-2.7.7.tar.gz -C /opt/

IMG_256

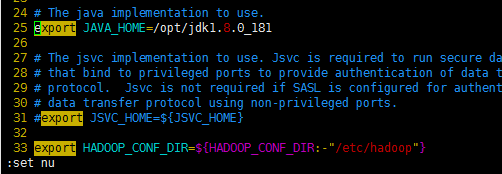
**2)、进入配置文件目录:cd /opt/hadoop-2.7.7/etc/hadoop/**

配置hadoop-env.sh，命令：

vim hadoop-env.sh #在hadoop-env.sh中修改JAVA\_HOME

25 export JAVA\_HOME=/usr/local/jdk #根据rpm -ql java-1.8.0-openjdk-devel 名称查看实际路径

33 export HADOOP\_CONF\_DIR="/usr/local/hadoop/etc/hadoop"



**3）、配置core-site.xml**

vim core-site.xml

#在文件中加入如下内  
<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://namenode01:9000</value>

</property>

<property>

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

<value>**/data/hadoop277/hadoop**</value> #数据保存路径

</property>

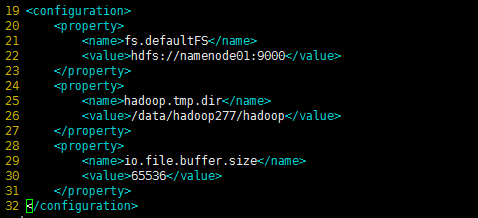
<property>

<name>io.file.buffer.size</name>

<value>65536</value>

</property>

</configuration>



**4）、修改hdfs-site.xml**

vim hdfs-site.xml

<configuration>

20 <property>

21 <name>dfs.replication</name>

22 <value>3</value>

23 </property>

24 <property>

25 <name>dfs.namenode.secondary.http-address</name>

26 <value>namenode01:50090</value>

27 </property>

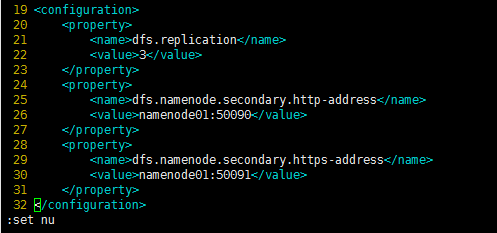
28 <property>

29 <name>dfs.namenode.secondary.https-address</name>

30 <value>namenode01:50091</value>

31 </property>

</configuration>



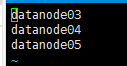
**5）、配置slaves文件**

vim slaves

datanode03

datanode04

datanode05  **# 添加数据节点的主机名**



**6）、配置yarn-site.xml**

vim yarn-site.xml

15 <configuration>

16 <property>

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

18 <value>namenode01</value>

19 </property>

20 <property>

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

22 <value>mapreduce\_shuffle</value>

23 </property>

24 <property>

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

26 <value>false</value>

27 <description>Whether virtual memory limits will be enforced for containers</description>

28 </property>

29 <property>

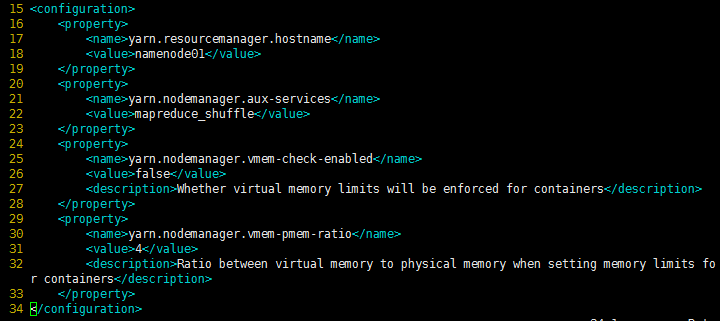
30 <name>yarn.nodemanager.vmem-pmem-ratio</name>

31 <value>4</value>

32 <description>Ratio between virtual memory to physical memory when setting memory limits fo r containers</description>

33 </property>

34 </configuration>



**7）、配置mapred-site.xml文件**

该文件是不存在的，把mapred-site.xml.template文件改名

cp mapred-site.xml.template mapred-site.xml

vim mapred-site.xml

19 <configuration>

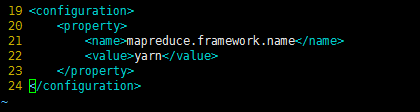
20 <property>

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

22 <value>yarn</value>

23 </property>

24 </configuration>





**8）、配置完以上几个配置文件后，可把配置文件复制到集群其它主机**

scp -r /opt/hadoop-2.7.7/ datanode03:/opt/hadoop-2.7.7

#datanode04和datanode05也是一样

**\*配置完成，接着格式化hdfs**

\*在主节点namenode01上执行以下命令

\*hdfs namenode -format（**只能操作一次，不能多次操作，公司集群已操作工，不需操作**）

**步骤三、在namenode01中启动HDFS和YARN**

（该步骤在namenode01中操作，其它主机查看服务即可）

**1）、在namenode01中启动HDFS**

201上有一个进程占用了9000端口：

lsof -i :9000 #查出进程号，然后kill掉。

没有这条命令，用yum -y install lsof安装

start-dfs.sh #启动hdfs服务

用命令jps查看各台机器上对应进程已经起来：

201主机：NameNode、SecondaryNameNode；

204-207主机：DataNode

**2）、启动Yarn**

在201上执行：

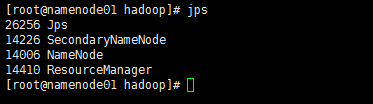
start-yarn.sh

jps查看各台机器上对应进程已经起来：

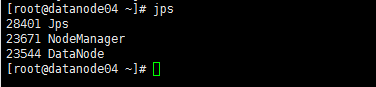
201主机新增服务：ResourceManager；

204-207新增服务：NodeManager；

201主机最终服务：

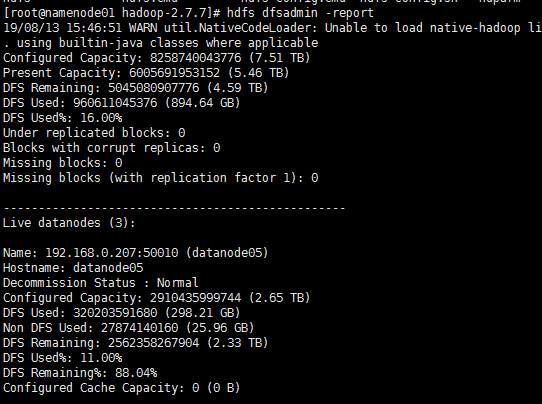


205、206、207主机最终服务：



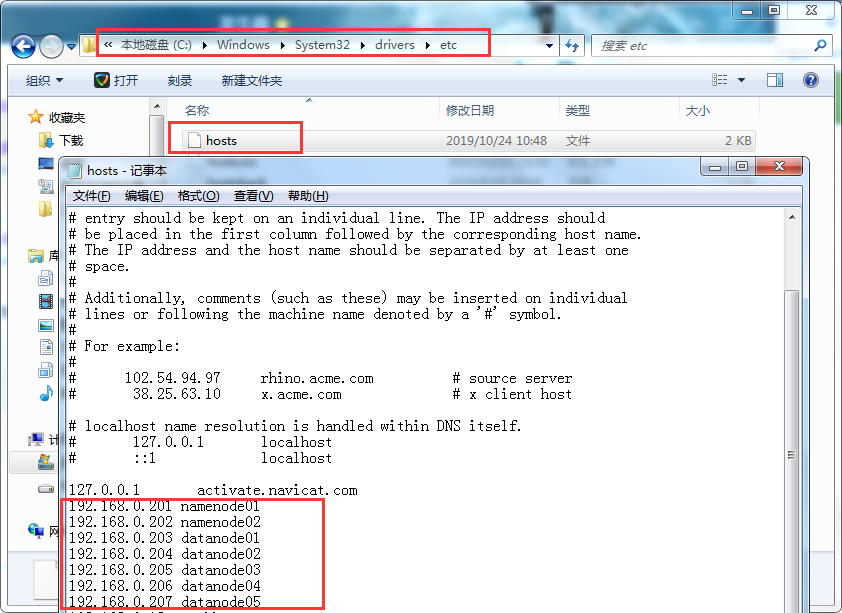
使用命令检测集群情况：

hdfs dfsadmin -report



hadoop jar hadoop-mapreduce-examples-2.5.0-cdh5.3.6.jar wordcount /test.txt /output  
　　　　　　　　　　　　　　注意:如果上面执行失败或者卡着不动，检查是否是进程宕掉了。  
　　　　　　　　　　　　　　[hadoop@slaver1 mapreduce]$ hadoop fs -cat /user/hadoop/output/part-r-00000查看统计的内容；

**Java连接hdfs操作：hdfs://192.168.0.201:9000**



Windows系统也可添加hosts文件进行域名解析，这样访问datanode01或者其它就可以对应ip地址

**可登陆hadoop网站检测集群情况：**

201主机

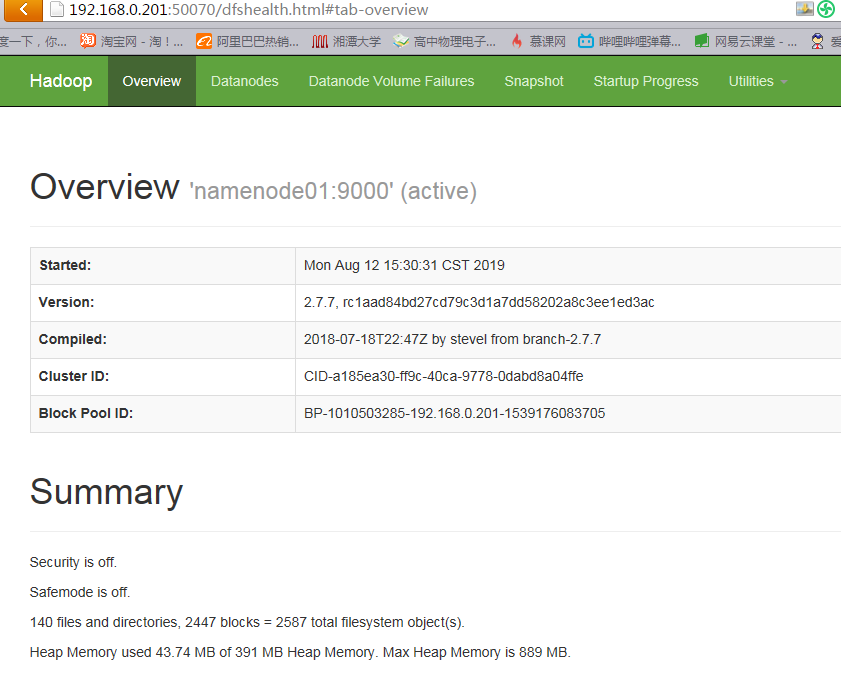
http://192.168.0.201:50070/ //--namenode web页面（namenode1）

http://192.168.0.201:50090/ //--secondory namenode web页面（namenode1）

http://192.168.0.201:8088/ //--secondory namenode web页面（namenode1）

http://192.168.0.205:50075/ //--datanode web页面（datanode3,datanode4,datanode5）

http://192.168.0.205:8042/ //--nodemanager web页面（datanode3,datanode4,datanode5）



**步骤四、配置**spark****

**1）、解压spark安装包到/usr/local/目录下**

命令：

tar -xvf spark-2.2.0-bin-hadoop2.6.tgz -C /opt

**2）、配置slaves**

命令：

cd  /opt/spark-2.3.2/conf  
cp slaves.template slaves

vim slaves

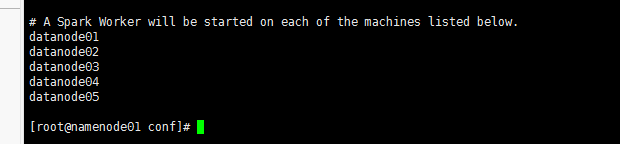
写入如下内容  
datanode01

datanode02

datanode03

datanode04

datanode05



**3）、配置spark-env.sh**

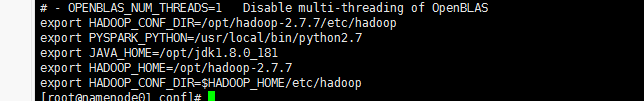
命令：

cp spark-env.sh.template spark-env.sh  
vi spark-env.sh

在文件中加入以下内容：

export JAVA\_HOME=/opt/jdk1.8.0\_181  
export HADOOP\_HOME=/opt/hadoop-2.7.7  
export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

如下：

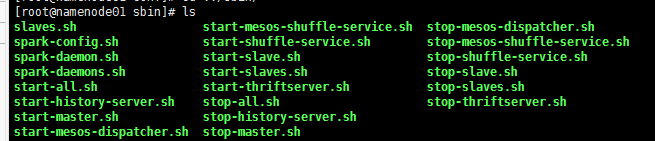


**把saprk目录拷贝到slaves文件添加的datanode节点中：**

**Scp spark-2.3.2 datanode01:/opt**

4）、启动SPARK：/opt/spark-2.3.2/sbin/start-all.sh

关闭SPARK：/opt/spark-2.3.2/sbin/stop-all.sh



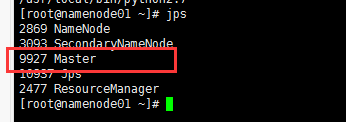
5）、验证有无启动

检测端口：netstat -anutlp | grep 8080



浏览器登录查看集群状态：



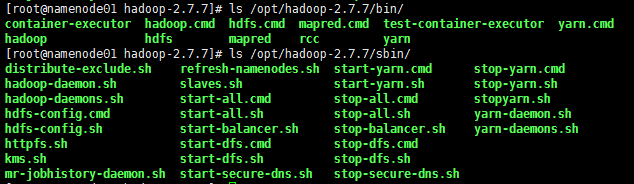


**步骤五、集群操作命令**

集群操作命令，在namenode01中执行（其它数据节点执行也可以），命令位置

/opt/hadoop-2.7.7/sbin/\*

/opt/hadoop-2.7.7/bin/\*

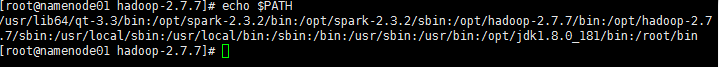


以上命令可以添加进bash中，以后就不需要输入绝对路径。

vim ~/.bashrc

#在最后一行添上：

export PATH=/opt/hadoop-2.7.7/sbin/:/opt/hadoop2.7.7/bin/:$PATH



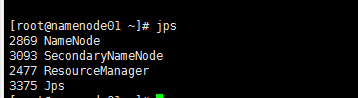
1. **、服务管理命令**

**在201上操作，即管理服务器**

start-all.sh stop-all.sh #开启和关闭所有集群服务，不推荐使用

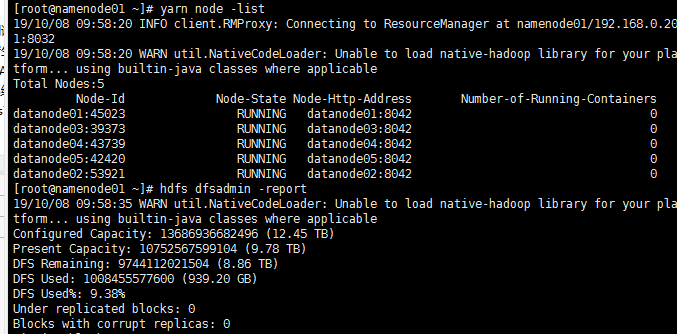
start-dfs.sh stop-dfs.sh #关闭hdfs服务，NameNode、DataNode和SecondaryNamenode会关闭

start-yarn.sh stop-yarn.sh #启动和关闭yarn服务，Resourcemanager和Nodemanager会关闭



yarn node -list #查看nodemanager运行情况

hdfs dfsadmin -report #产看datanode运行情况



1. **、集群文件管理及常用命令使用**

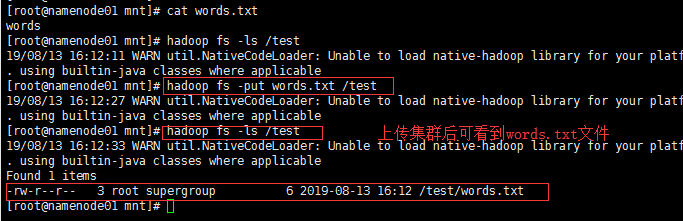
使用命令检测集群情况：

hdfs dfsadmin -report

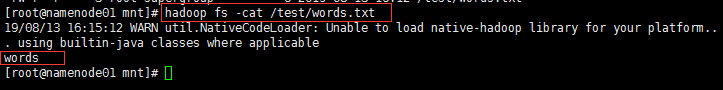
**hadoop fs -[命令] 目录**

**hdfs dfs -[命令] 目录**  #两条命令都是对文件管理,hadoop命令针对所有系统文件，使用最广，hdfs命令只能针对HDFS文件，以下案例以hadoop为例，hdfs命令也可得到类似结果

上传文件至集群：hadoop fs -put words.txt /test/ #把本地的words.txt文件上传至集群中的/test目录中



查看集群文件内容： **hadoop fs -cat /test/words**



从集群下载某个文件，下载到当前路径：

**hadoop fs -get /path/wc/input/words.txt**

显示一个文件的末尾

**hadoop fs -tail /weblog/test.log**

在hdfs上创建目录

**hadoop fs -mkdir -p /aaa/bbb/cc/dd**

从hdfs的一个路径拷贝hdfs的另一个路径

**hadoop fs -cp /aaa/test1.txt /bbb/test2.txt**

删除目录及其里面内容

**hadoop fs -rm -r /path/wc/input/words.txt**

与linux文件系统中的用法一样，对文件所属权限

**1）hadoop fs -chmod 666 /hello.txt**

**2）hadoop fs -chown someuser:somegrp /hello.txt**

统计文件系统的可用空间信息

**hadoop fs -df -h /**

统计文件夹的大小信息

**hadoop fs -du -s -h /aaa/\***

**五、节点管理-新增数据节点**

A：在新增节点修改主机名datanode02，安装jdk环境

B：Namenode01节点中设置到新节点的无密码登陆

C：在Namenode01节点slaves文件中添加新节点

D：在所有节点/etc/hosts文件中增加新节点（所有节点保持一致）

E：在新节点中执行

hadoop-daemon.sh start datanode

yarn-daemon.sh start nodemanager

start-balancer.sh #同步

01. [root@hadoop5 ~]# echo datanode02 > /etc/hostname //在datanode02中更改主机名为datanode02

02. [root@hadoop5 ~]# hostname datanode

03. [root@datanode02 ~]# yum -y install java-1.8.0-openjdk-devel

04. [root@dataanode02 ~]# mkdir /opt/hadoop

05. [root@namenode01 .ssh]# ssh-copy-id 192.168.0.204 #在namenode01中操作

06. [root@namenode01 .ssh]# vim /etc/hosts

07. 192.168.0.201 namenode01

08. 192.168.0.204 node1

09. 192.168.0.205 node2

10. 192.168.0.206 node3

11. 192.168.0.207 node4

12. [root@namenode01 .ssh]# scp /etc/hosts 192.168.1.65:/etc/

13. [root@namenode01 ~]# cd /opt/hadoop-2.7.7/

14. [root@namenode01 hadoop-2.7.7]# vim ./etc/hadoop/slaves

15. datanode02

16. datanode03

17. datanode04

18. datanode05

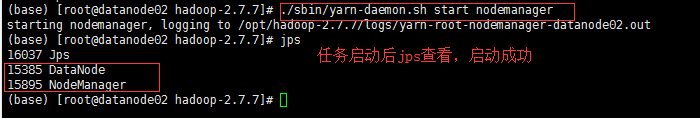
19. [root@namenode01 hadoop-2.7.7]# for i in {62..65}; do rsync -aSH --delete /usr/local/hadoop/

20. \ 192.168.1.$i:/opt/hadoop-2.7.7/ -e 'ssh' & done //同步配置

25. [root@datanode02 ~]# cd /opt/hadoop-2.7.7/

26. [root@datanode02 hadoop-2.7.7]# ./sbin/hadoop-daemon.sh start datanode

[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh start nodemanager #在datanode02中单独启动，然后用jps查看状态



在新的节点上执行：  
　　　　　　hadoop-daemon.sh start tasktracker

均衡之前的datanode的block，这个会非常耗时   
1）如果不balance，那么cluster会把新的数据都存放在新的node上，这样会降低mapred的工作效率   
2）设置平衡阈值，默认是10%，值越低各节点越平衡，但消耗时间也更长   
[root@slave-004 hadoop]# start-balancer.sh -threshold 5   
3）设置balance的带宽，默认只有1M/s，修改hdfs-site.xml

　　　　　<property>   
　　　　　　　<name>dfs.balance.bandwidthPerSec</name>   
　　　　　　　<value>1048576</value>   
　　　　　</property>

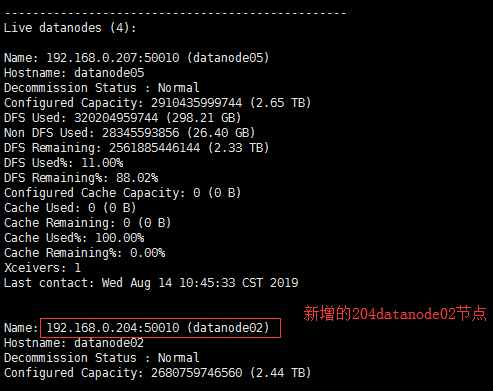
**设置同步带宽，在新的节点上执行：**

01. root@datanode02 hadoop-2.7.7]# ./bin/hdfs dfsadmin -setBalancerBandwidth 6000000 #6M/s

02. Balancer bandwidth is set to 60000000

03. root@datanode02 hadoop-2.7.7]# ./sbin/start-balancer.sh #开始同步

在namenode01主机中查看节点状态：hdfs dfsadmin -report



**六、节点管理-删除数据节点**

01. [root@namenode01 hadoop-2.7.7]# vim/opt/hadoop-2.7.7/etc/hadoop/slaves

02. #去掉之前添加的datanode02

03.datanode03

04. datanode04

05. datanode05

06. [root@namenode01 hadoop-2.7.7]# vim /opt/hadoop-2.7.7/etc/hadoop/hdfs-site.xml

07. #在此配置文件里面加入下面四行

08. <property>

09. <name>dfs.hosts.exclude</name>

10. <value>/opt/hadoop-2.7.7/etc/hadoop/exclude</value>

11. </property>

12.

13. [root@namenode01 hadoop-2.7.7]# vim /opt/hadoop-2.7.7/etc/hadoop/exclude datanode02

**导出数据**

01. [root@namenode01 hadoop-2.7.7]# ./bin/hdfs dfsadmin -refreshNodes

02. Refresh nodes successful

03. [root@namenode01 hadoop-2.7.7]# ./bin/hdfs dfsadmin -report #查看datanode02显示Decommissioned

04.

05. Name: 192.168.0.203:50010 (datanode02)

06. Hostname: datanode02

07. Decommission Status : Decommissioned

08. Configured Capacity: 2135949312 (1.99 GB)

09. DFS Used: 4096 (4 KB)

1. Non DFS Used: 1861509120 (1.73 GB)

[root@datanode02 hadoop-2.7.7]# ./sbin/hadoop-daemon.sh stop datanode //停止datanode

stopping datanode

[[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh start nodemanager

//yarn 增加 nodemanager

[root@datanode02 hadoop-2.7.7]# ./sbin/yarn-daemon.sh stop nodemanager //停止nodemanager

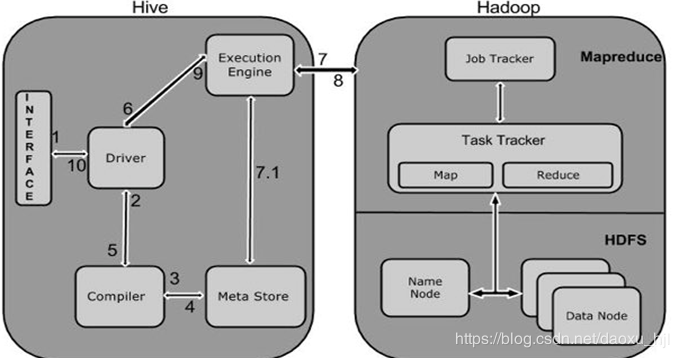
stopping nodemanager

[root@datanode02 hadoop-2.7.7]# ./bin/yarn node -list

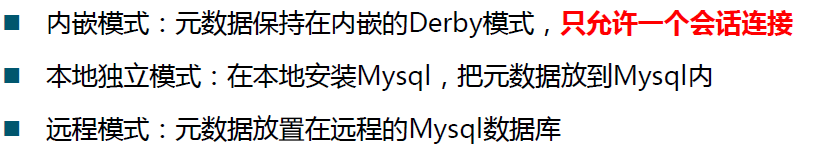
//yarn 查看节点状态，还是有node4节点，要过一段时间才会消失

# 安装配置hive

Hive工作原理：



安装模式：



Hive只需要一个节点安装，不需要每个节点安装，**本次hive在201上搭建；本次搭建需要安装mysql，命令：**

**yum -y install mysql mysql-server mysql-devel**

**启动 service mysqld start**

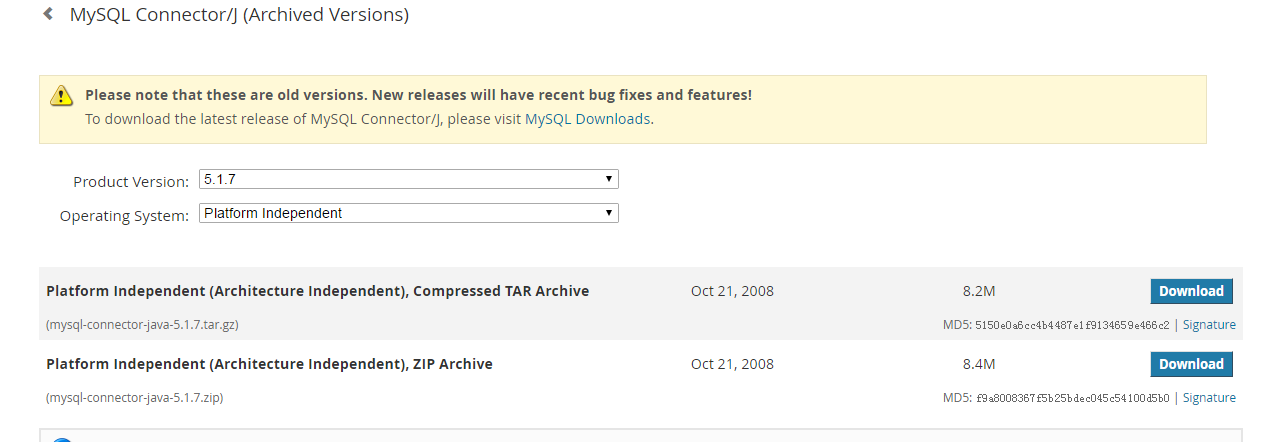
**授权账户和权限：**

**grant all on \*.\* to root@’%’ identified by ‘wislife123’;**

依赖软件：

     \*Hive工具使用JDBC方式连接MySQL数据库，需要用到MySQL数据库连接工具软件，选用数据库相应的版本号，201上数据库的版本号为mysql-5.1.73，驱动程序软件包名：mysql-connector-java-5.1.7-bin.jar；

下载地址：<https://downloads.mysql.com/archives/c-j/>

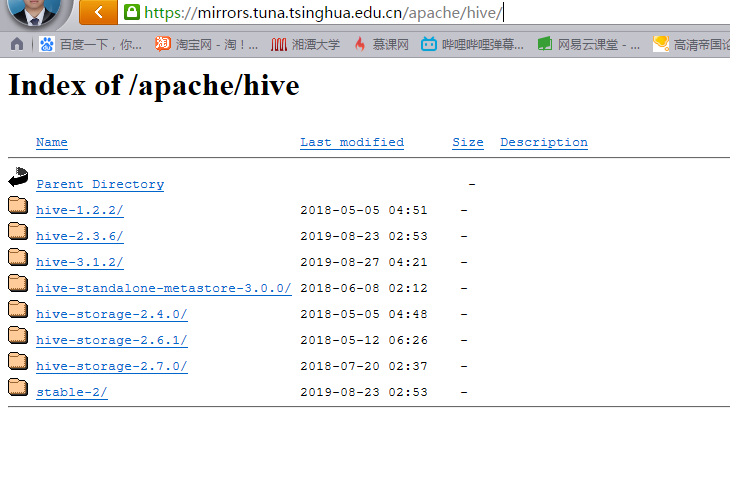


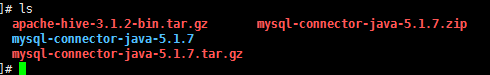
下载后解压软件包，找到mysql-connector-java-5.1.7-bin.jar，拷贝到hive软件包lib目录下： cp mysql-connector-java-5.1.7-bin.jar /opt/apache-hive-3.1.2-bin/lib

1. .hive下载地址

下载地址：<https://mirrors.tuna.tsinghua.edu.cn/apache/hive/>

本次部署版本为：hive-3.1.2

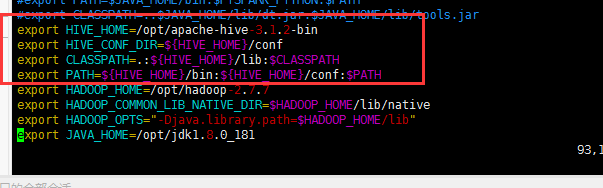




201服务器上下载了apache-hive-3.1.2-bin.tar.gz，解压后把软件包放在:/opt/apache-hive-3.1.2-bin/ 目录中

1. .设置环境变量

vim /etc/profile



让配置生效：source /etc/profile

Hive客户端工具后续将使用Beeline 替代HiveCLI ，并且后续版本也会废弃掉HiveCLI 客户端工具。

1. .修改配置文件hive-site.xml

在/opt/apache-hive-3.1.2-bin/conf目录下,生成配置文件：

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

修改配置文件：vim hive-site.xml

#修改 属性：hive.metastore.warehouse.dir 为 /data/hive/warehouse  
#修改 属性：hive.exec.scratchdir 为 /data/hive/tmp

#创建warehouse目录

hdfs dfs -mkdir -p /data/hive/warehouse

hdfs dfs -chmod 777 /data/hive/warehouse

#创建tmp目录

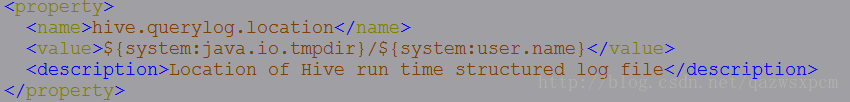
hdfs dfs -mkdir -p /data/hive/tmp

hdfs dfs -chmod 777 /data/hive/tmp

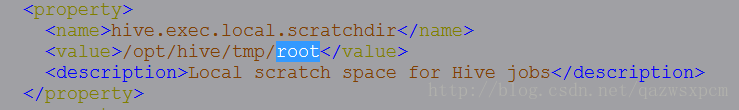
将配置文件中所有的${system:java.io.tmpdir},改为 /opt/hive/tmp (如果没有该文件则创建)或者/opt/apache-hive-3.1.2-bin/tmp，并将此文件夹赋予读写权限：chmod 777 /opt/hive/tmp;

将${system:user.name}更改为 root

更改前：



更改后：



Cp $HIVE\_HOME/conf/hive-env.sh.template $HIVE\_HOME/conf/hive-env.sh

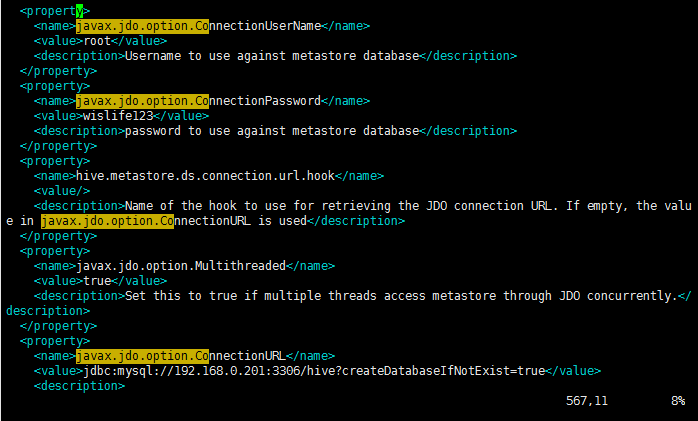
vim $HIVE\_HOME/conf/hive-env.sh  
增加如下内容

export HADOOP\_HOME=/opt/hadoop-3.2.1

export HIVE\_CONF\_DIR=/opt/apache-hive-3.1.2-bin/conf

export HIVE\_AUX\_JARS\_PATH=/opt/apache-hive-3.1.2-bin/lib

hive 默认为 derby 数据库， derby 数据只运行单个用户进行连接，所以这里需要调整为 Mysql 数据库，以下为修改配置内容：



Connectionusername：root ConnectionPasswd：wislife123 #mysql数据库连接名和密码

ConncetionURL：jdbc:mysql://192.168.0.201:3306/hive?createDatabaseIfNotExist=true

ConnectionDriverName:com.mysql.jdbc.Driver

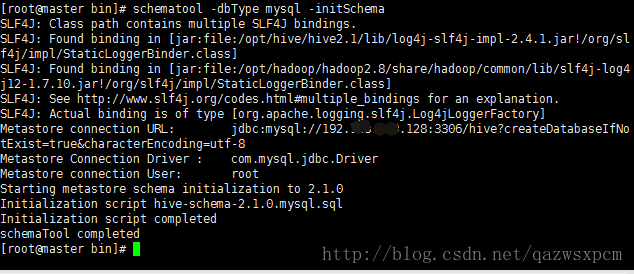
jdbc:derby:;databaseName=metastore\_db;create=true

org.apache.derby.jdbc.EmbeddedDriver

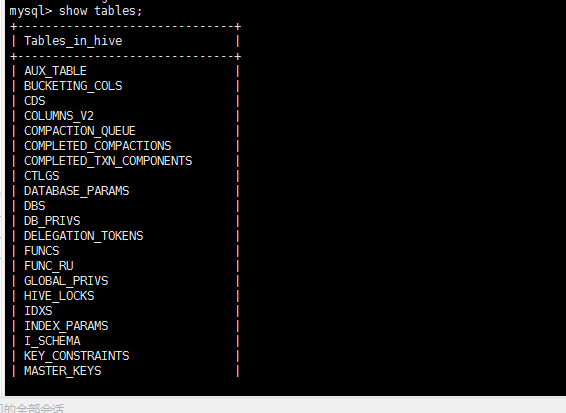
首先初始化数据库,初始化的时候注意要将mysql启动,输入:

schematool -initSchema -dbType mysql

注：若上一步的hive-site.xml配置文件没配置对或缺少mysql的驱动程序jar包，会连接不上，而且初始化只能执行一次，执行多次会报数据库表已存在，初始化失败。

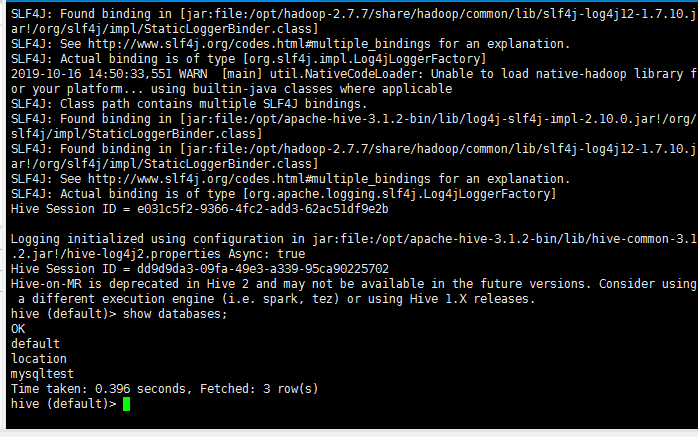


执行成功之后，可以看到hive数据库和一堆表已经创建成功了:



此时可用hive命令直接连接hive操作：

hive



导出内容，而且是非交互式的操作

Hive -e “select \* from 表 where 条件” >> 目录

Beeline是Hive新的命令行客户端工具。

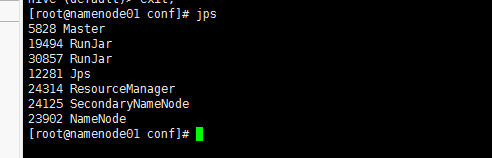
HiveServer2 支持一个新的命令行Shell，称为Beeline，它是基于SQLLine CLI的JDBC客户端。

Beeline支持嵌入模式(embedded mode)和远程模式(remote mode)。在嵌入式模式下，运行嵌入式的Hive(类似Hive CLI)，而远程模式可以通过Thrift连接到独立的HiveServer2进程上

Hive启动：

nohup hive --service metastore & #在 201 节点上运行 jps 应该会有RunJar 进程则启动成功

nohup hive --service hiveserver2 & #在 201 节点上运行 jps 应该会有Master 进程则启动成功



1. ##此处hs2的默认端口号为10000,可修改，hadoop001为主机名，hwzhdb为你连上hive后要进入哪个数据库，然后使用-n 指定当前用户名，其它则为固定写法。\*
2. [hadoop@hadoop001 bin]$ ./beeline -u jdbc:hive2://hadoop001:10000/hwzhdb -n hadoop
3. 201上命令：./beeline -u jdbc:hive2://namenode01:10000/location -n hadoop

密码123456

## 连接数据库：

            1    你可以在交互式模式下输入hiveserver连接字符串

            2    你也可以在CLI直接输入hiveserver连接字符串，beeline -u xxx -n yyy

## beeline连接选项

-u <database URL> JDBC连接串，beeline -u db\_URL

-r 连接到用户最后一次连接的hiveserver上，前提是用户使用的 !connect去连接的hiveserver，同时使用 !save保存了连接， beeline -r

-n <username> 连接的用户名，beeline -n valid\_user

-p <password> 连接的密码，beeline -p valid\_password -d <driver class> 使用哪个driver连接hiveserver，beeline -d driver\_class

-e <query> 用单引号或者双引号括起来的查询语句，beeline -e "query\_string"

-f <file> 要执行的脚本，beeline -f filepath

-i (or) --init <file or files> 初始化文件， beeline -i /tmp/initfile

-w (or) --password-file <password file> 从一个文件读取密码

-a (or) --authType <auth type> jdbc连接时的认证类型

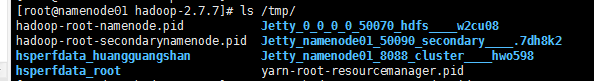
--property-file <file> 读取配置文件，beeline

--property-file /tmp/a

--hiveconf property=value 设置hiveconf参数，beeline --hiveconf prop1=value1 --hivevar name=value 设置hivevar参数，beeline --hivevar var1=value1 --color=[true/false] 终端显示是否带颜色，beeline --color=true --showHeader=[true/false] 查询结果中是否显示表字段名，beeline --showHeader=false --headerInterval=ROWS 重新显示列标题的间隔，以outputformat为表格的行数。 默认值为100， beeline --headerInterval=50 --autoCommit=[true/false] 是否开启事物的自动提交，默认false --showWarnings=[true/false] 是CLI使用HQL的时候是否显示warning信息 --showDbInPrompt=[true/false] 是否在命令行提示符显示数据库名，默认false，beeline --showDbInPrompt=true --showNestedErrs=[true/false] 是否显示嵌套错误，默认false. --force=[true/false] 当执行脚本发生错误时，是否强制执行，默认false --autosave=[true/false] 是否自动保存首选项，默认false，beeline --autosave=true --outputformat=[table/vertical/csv/tsv/dsv/csv2/tsv2] 查询显示的输出格式 --delimiterForDSV= DELIMITER 查询输出值的默认分隔符，默认是 '|'

**八、更改hadoop管理服务器的pid文件路径**

注：不更改pid路径，dfs和yarn的pid默认会存放在/tmp目录下



该目录的文件是临时文件，超过30天会自动清空，pid文件消失后，先关闭yarn和dfs就关闭不了，所以要更改pid文件路径

在管理服务器192.168.0.201上操作

[hadoop@hadoop000 ~]$ mkdir -p /opt/hadoop-2.7.7/pids

[hadoop@hadoop000 ~]$ jps

10512 ResourceManager

8018 Jps

10004 SecondaryNameNode

9655 NameNode

9806 DataNode

10623 NodeManager

[hadoop@hadoop000 ~]$ cd /opt/hadoop-2.7.7/sbin/

[hadoop@hadoop000 sbin]$ ./stop-dfs.sh

[hadoop@hadoop000 sbin]$ ./stop-yarn.sh

[hadoop@hadoop000 sbin]$ vi hadoop-daemon.sh

HADOOP\_PID\_DIR=/opt/hadoop-2.7.7/pids #新增这一行

[hadoop@hadoop000 sbin]$ vi yarn-daemon.sh

YARN\_PID\_DIR=/opt/hadoop-2.7.7/pids

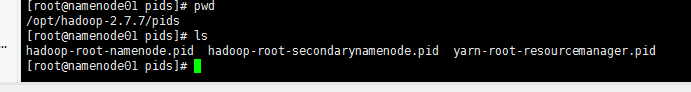
[hadoop@hadoop000 sbin]$ ./start-dfs.sh

[hadoop@hadoop000 sbin]$ ./start-yarn.sh

验证

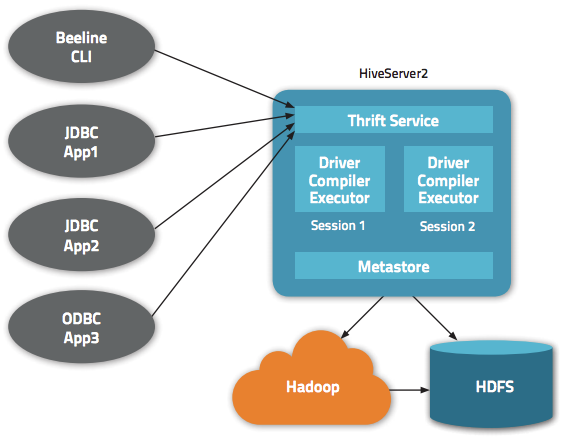
Jps验证服务有没启动，查看目录

ls /opt/hadoop-2.7.7/pids



**--制作人 刘兆臣**

**时间：2019.08.13**



[Hive](https://www.iteblog.com/archives/tag/hive/" \o "" \t "https://www.iteblog.com/archives/_blank) 除了为我们提供一个 CLI 方式来查询数据之外，还给我们提供了基于 JDBC/ODBC 的方式来连接 [Hive](https://www.iteblog.com/archives/tag/hive/" \o "" \t "https://www.iteblog.com/archives/_blank)，这就是 HiveServer2（HiveServer）。但是默认情况下通过 JDBC 连接 HiveServer2 不需要任何的权限认证（hive.server2.authentication = NONE）；这意味着任何知道 ThriftServer 地址的人都可以连接我们的 Hive，并执行一些操作。这些人甚至可以执行一些诸如 drop table xxx 之类的操作，这势必威胁到我们的数据安全。

HiveServer2 支持多种用户安全认证方式：NONE, NOSASL, KERBEROS, LDAP, PAM ,CUSTOM 等等。我们可以通过 hive.server2.authentication 参数进行配置。这篇文章我们涉及到的配置是

Hive启动命令：nohup hive --service metastore &

nohup hive --service hiveserver2&