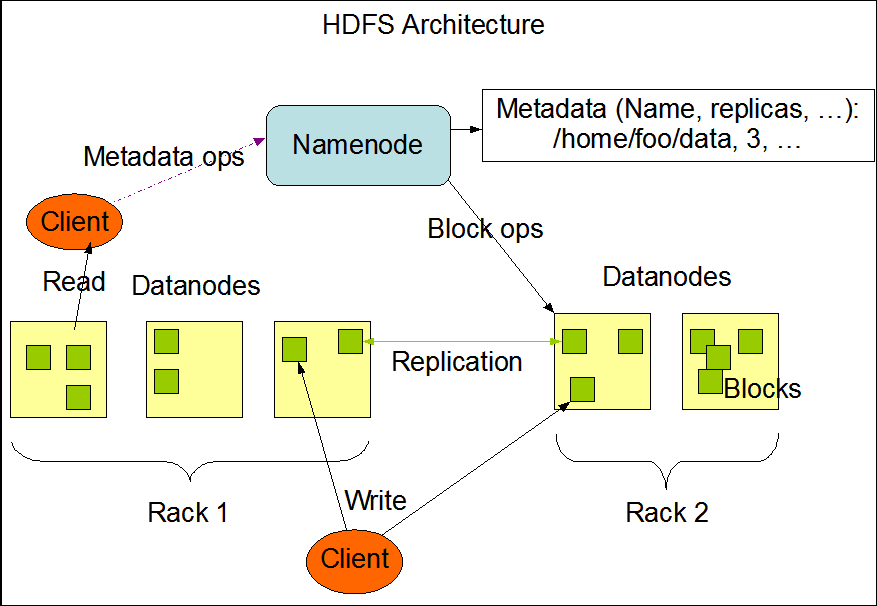
Hadoop集群部署

[Hadoop](http://www.linuxidc.com/topicnews.aspx?tid=13)是一个由Apache基金会所开发的分布式系统基础架构，Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序；HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

  
**HDFS架构图**

## Hadoop框架简介

Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

HDFS(Hadoop Distribution File System)，称为Hadoop分布式文件系统，主要特点：

* HDFS最小以64MB的数据块存储文件，相比其他文件系统中的4KB~32KB分块大得多。
* HDFS在时延的基础上对吞吐量进行了优化，它能够高效处理了对大文件的读请求流，但不擅长对众多小文件的定位请求
* HDFS对普通的“一次写入，多次读取”的工作负载进行了优化。
* 每个存储节点运行着一个称为DataNode的进程，它管理着相应主机上的所有数据块。这些存储节点都由一个称为NameNode的主进程来协调，该进程运行于一台独立进程上。
* 与磁盘阵列中设置物理冗余来处理磁盘故障或类似策略不同，HDFS使用副本来处理故障，每个由文件组成的数据块存储在集群众的多个节点，HDFS的NameNode不断监视各个DataNode发来的报告。

### MapReduce工作原理

客户端，提交MapReduce作业；jobtracker，协调作业的运行，jobtracker是一个java应用程序，它的主类是JobTracker；tasktracker。运行作业划分后的任务，tasktracker是一个java应用程序，TaskTracker是主类。

### Hadoop优点

Hadoop是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在Hadoop上开发和运行处理海量数据的应用程序。它主要有以下几个优点：

高可靠性：Hadoop按位存储和处理数据的能力值得人们信赖。

高扩展性：Hadoop是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中。

高效性：Hadoop能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此处理速度非常快。

高容错性：Hadoop能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

低成本：与一体机、商用数据仓库以及QlikView、Yonghong Z-Suite等数据集市相比，hadoop是开源的，项目的软件成本因此会大大降低。

Hadoop带有用Java语言编写的框架，因此运行在 Linux 生产平台上是非常理想的。Hadoop 上的应用程序也可以使用其他语言编写，比如 C++。

Hadoop官网：http://hadoop.apache.org/

## 先决条件

保持Hadoop集群每个节点配置环境一致，安装java，配置ssh。

实验环境：

Platform：VMware 10

OS: [CentOS](http://www.linuxidc.com/topicnews.aspx?tid=14) 6.7

Software: hadoop-2.6.5.tar.gz, jdk-8u151-linux-x64.tar.gz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hostname | IP Address | OS version | Hadoop role | Node role |
| master | 192.168.88.88 | CentOS 6.7 | Master | namenode |
| slave1 | 192.168.88.89 | CentOS 6.7 | Slave | datenode |
| slave2 | 192.168.88.90 | CentOS 6.7 | Slave | datenode |

#把需要的软件包下载下来上传到集群的各个节点上

## 集群的构架和安装

### Hosts文件设置

#修改hosts文件，需在hadoop集群的每个节点都操作

[root@master ~]# cat /etc/hosts

192.168.88.88 master

192.168.88.89 slave1

192.168.88.90 slave2

### 安装java

#略

### 安装hadoop

#创建hadoop用户，设置使用sudo，需在hadoop集群的每个节点都操作

[root@master ~]# useradd hadoop && echo 123456 | passwd --stdin hadoop

[root@master ~]# echo "hadoop ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers

#上传安装包

[root@master ~]# su - hadoop

[hadoop@master ~]$ cd /tmp/pkg

[hadoop@master pkg]$ sudo rz # 上传hadoop-2.6.5.tar.gz

[hadoop@master pkg]$ sudo mkdir /opt/bigdata

[hadoop@master pkg]$ sudo tar -zxvf hadoop-2.6.5.tar.gz -C /opt/bigdata/ && cd /opt/bigdata/

[hadoop@master bigdata]$ sudo mv hadoop-2.6.5/ hadoop

[hadoop@master bigdata]$ sudo chown -R hadoop:hadoop hadoop/

#将hadoop的二进制目录添加到PATH变量，并设置HADOOP\_HOME环境变量，需在hadoop集群的每个节点都操作

[hadoop@master bigdata]$ egrep HADOOP\_HOME /etc/profile

export HADOOP\_HOME=/opt/bigdata/hadoop

export PATH=$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

[hadoop@master bigdata]$ source /etc/profile

### 创建hadoop相关目录

#创建节点存储目录，集群的每个节点都操作

sudo mkdir -p /data/hdfs/{name,data}

sudo chown -R hadoop:hadoop /data/

#节点存储备份目录，集群的每个节点都操作

sudo mkdir -p /data/hdfs\_tmp/{name,data}

### SSH配置

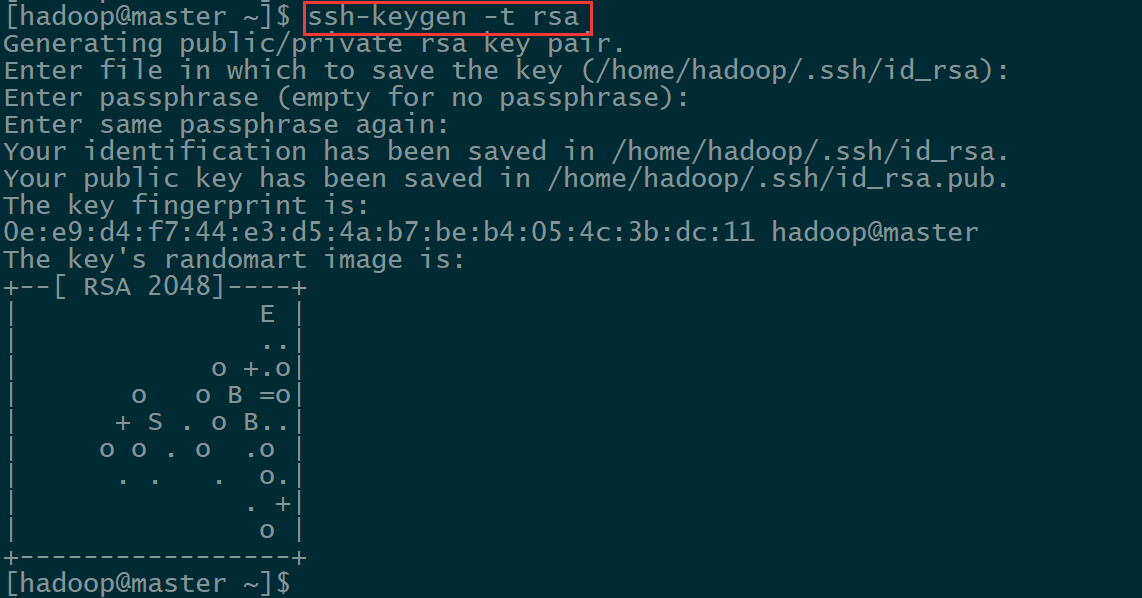
#设置集群主节点免密码登陆其他节点

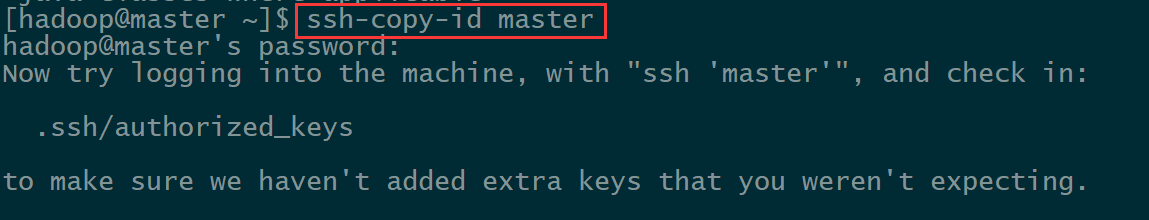
[hadoop@master ~]$ ssh-keygen -t rsa

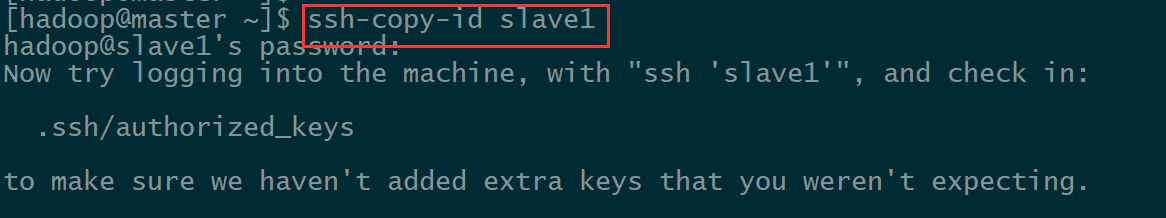
[hadoop@master ~]$ ssh-copy-id master

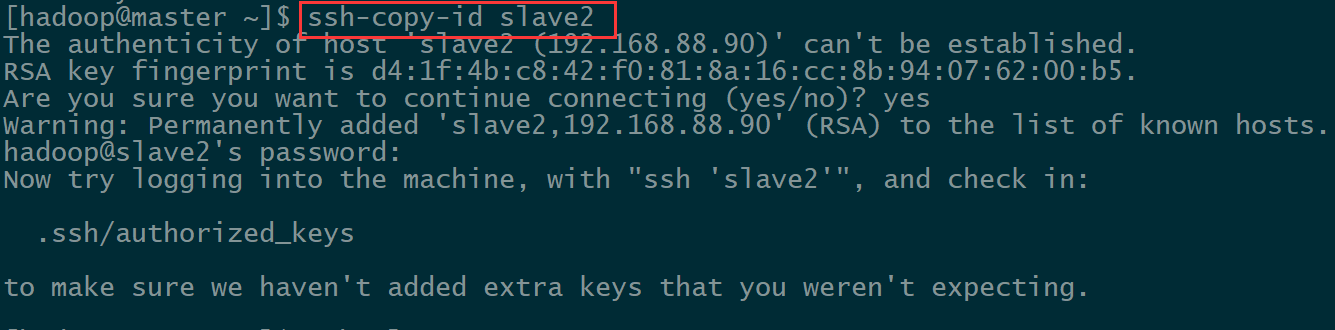
[hadoop@master ~]$ ssh-copy-id slave1

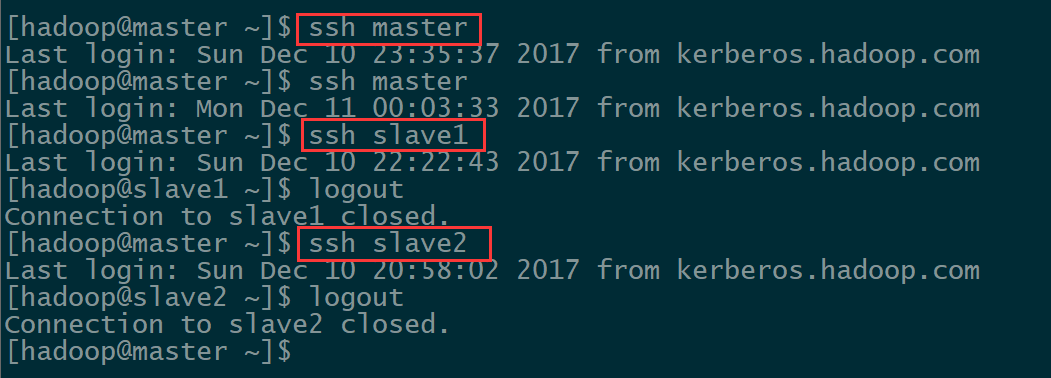
[hadoop@master ~]$ ssh-copy-id slave2









#测试ssh登录  


### 修改hadoop的配置文件

文件位置：/opt/bigdata/hadoop/etc/hadoop，文件名称：hadoop-env.sh、yarn-evn.sh、slaves、core-site.xml、hdfs-site.xml、mapred-site.xml、 yarn-site.xml

（1）配置hadoop-env.sh文件

#在hadoop安装路径下，进入hadoop/etc/hadoop/目录并编辑hadoop-env.sh，修改JAVA\_HOME为JAVA的安装路径

[hadoop@master bigdata]$ cd /opt/bigdata/hadoop/etc/hadoop/

[hadoop@master hadoop]$ egrep JAVA\_HOME hadoop-env.sh

# The only required environment variable is JAVA\_HOME. All others are

# set JAVA\_HOME in this file, so that it is correctly defined on

#export JAVA\_HOME=${JAVA\_HOME}

export JAVA\_HOME=/opt/jdk1.8.0\_151/

（2）配置yarn.sh文件

指定yarn框架的java运行环境，该文件是yarn框架运行环境的配置文件，需要修改JAVA\_HOME的位置。

[hadoop@master hadoop]$ grep JAVA\_HOME yarn-env.sh

# export JAVA\_HOME=/home/y/libexec/jdk1.6.0/

export JAVA\_HOME=/opt/jdk1.8.0\_151/

if [ "$JAVA\_HOME" != "" ]; then

#echo "run java in $JAVA\_HOME"

JAVA\_HOME=$JAVA\_HOME

if [ "$JAVA\_HOME" = "" ]; then

echo "Error: JAVA\_HOME is not set."

JAVA=$JAVA\_HOME/bin/java

（3）配置slaves文件

指定DataNode数据存储服务器，将所有的DataNode的机器的主机名写入到此文件中，如下：

[hadoop@master hadoop]$ cat slaves

slave1

slave2

Hadoop 3种运行模式

本地独立模式：Hadoop的所有组件，如NameNode，DataNode，Jobtracker，Tasktracker都运行在一个java进程中。

伪分布式模式：Hadoop的各个组件都拥有一个单独的Java虚拟机，它们之间通过网络套接字通信

完全分布式模式：Hadoop分布在多台主机上，不同的组件根据工作性质的不同安装在不通的Guest上。

#配置完全分布式模式

（4）修改core-site.xml文件，添加红色区域的代码

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://master:9000</value>

</property>

<property>

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

<value>131072</value>

</property>

<property>

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

<value>/data/hdfs\_tmp/</value>

<description>Abase for other temporary directories.</description>

</property>

</configuration>

（5）修改hdfs-site.xml文件，灰色部份为SecondNameNode无需配置，后面将配置HA机制，替代SecondNameNode

<configuration>

<property>

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

<value>master:9001</value>

<description># 通过web界面来查看HDFS状态 </description>

</property>

<property>

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

<value>/data/hdfs/name</value>

</property>

<property>

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

<value>/data/hdfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

<description># 每个Block有2个备份</description>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

（6）修改mapred-site.xml

这个是mapreduce任务的配置，由于hadoop2.x使用了yarn框架，所以要实现分布式部署，必须在mapreduce.framework.name属性下配置为yarn。mapred.map.tasks和mapred.reduce.tasks分别为map和reduce的任务数。

[hadoop@master hadoop]$ mv mapred-site.xml.template mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

<property>

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

<value>master:10020</value>

</property>

<property>

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

<value>master:19888</value>

</property>

</configuration>

（7）配置节点yarn-site.xml

#该文件为yarn架构的相关配置

mapred-site.xml

<?xml version="1.0"?>

<!-- mapred-site.xml  -->

<configuration>

<property>

<name>mapred.child.java.opts</name>

<value>-Xmx400m</value>

<!--Not marked as final so jobs can include JVM debuggung options -->

</property>

</configuration>

yarn-site.xml

<?xml version="1.0"?>

<!-- yarn-site.xml  -->

<configuration>

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

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

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

<value>master:8032</value>

</property>

<property>

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

<value>master:8030</value>

</property>

<property>

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

<value>master:8031</value>

</property>

<property>

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

<value>master:8033</value>

</property>

<property>

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

<value>master:8088</value>

</property>

<property>

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

<value>8192</value>

</property>

</configuration>

### hadoop 日志配置

日志路径：${HADOOP\_HOME}/logs，可配置到公共路径下。

### 复制hadoop到其他节点

在slave1和slave2节点

[hadoop@slave1 opt]$ sudo mkdir bigdata

[hadoop@slave1 opt]$ sudo chown -R hadoop:hadoop bigdata/

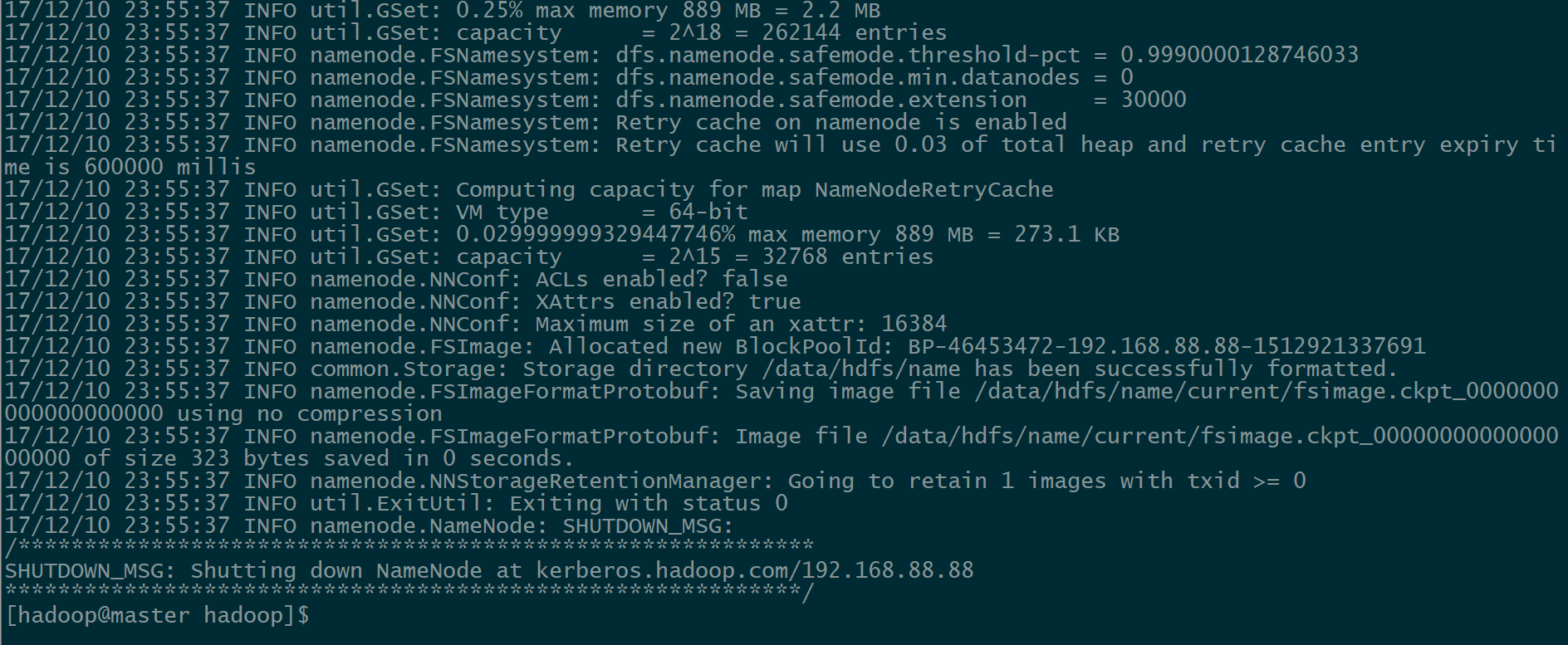
在master节点

[hadoop@master hadoop]$ scp -r /opt/bigdata/hadoop/ slave1:/opt/bigdata/

[hadoop@master hadoop]$ scp -r /opt/bigdata/hadoop/ slave2:/opt/bigdata/

### 在master使用hadoop用户初始化NameNode

[hadoop@master hadoop]$ hdfs namenode -format

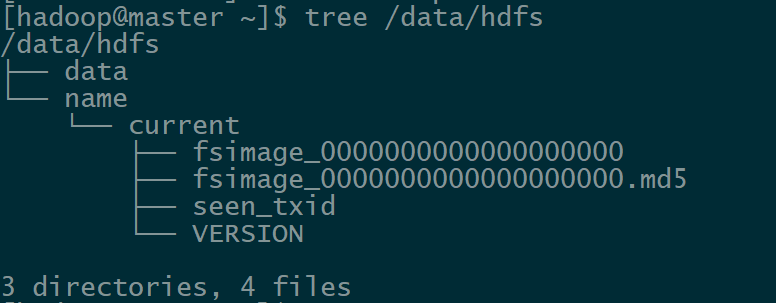


[hadoop@master hadoop]$ su -

[root@master ~]# yum -y install tree

[root@master ~]# su – hadoop

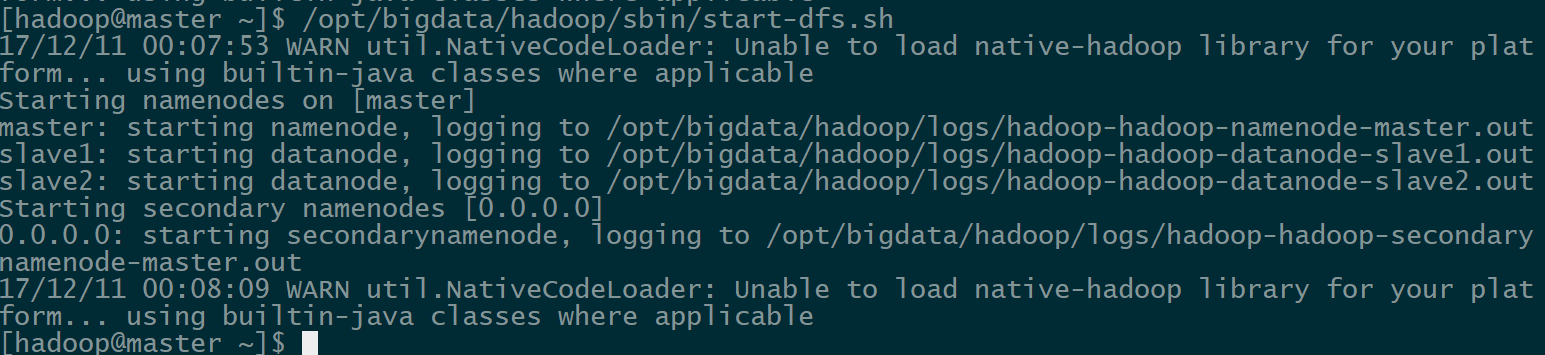
[hadoop@master ~]$ tree /data/hdfs



### 启动hadoop

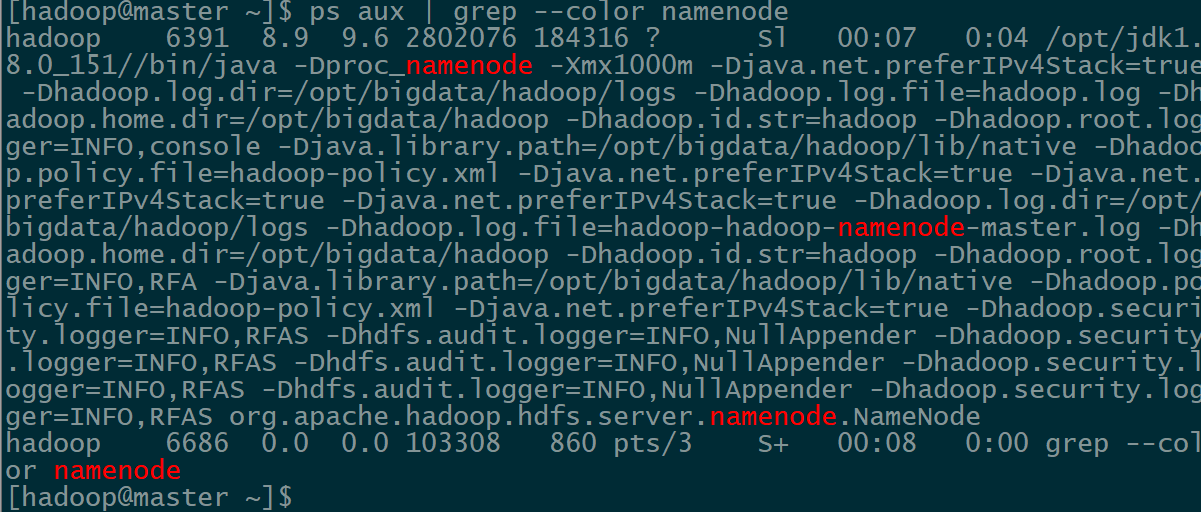
[hadoop@master ~]$ start-dfs.sh

[hadoop@master ~]$ stop-dfs.sh



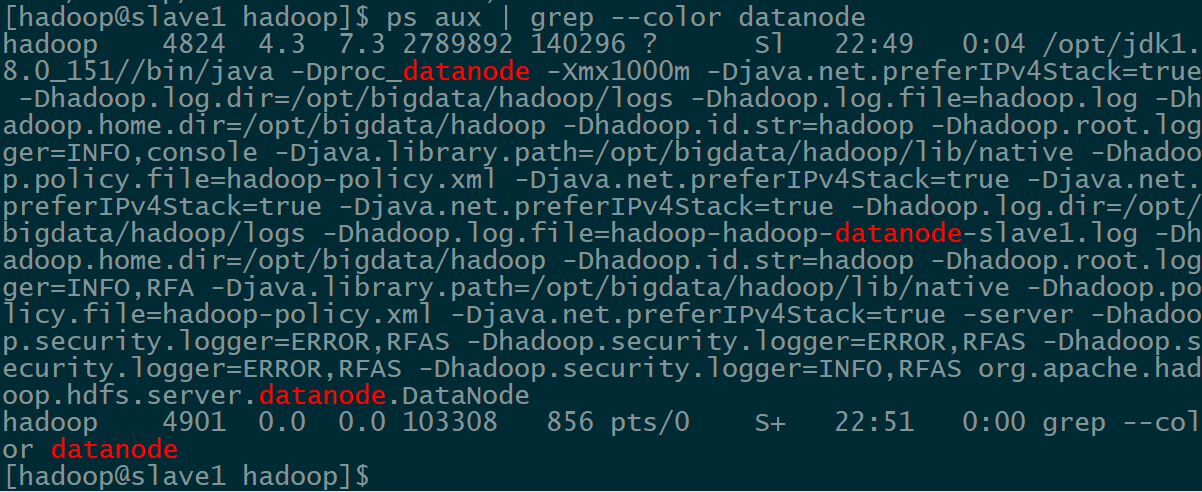
#namenode节点（master）上面查看进程

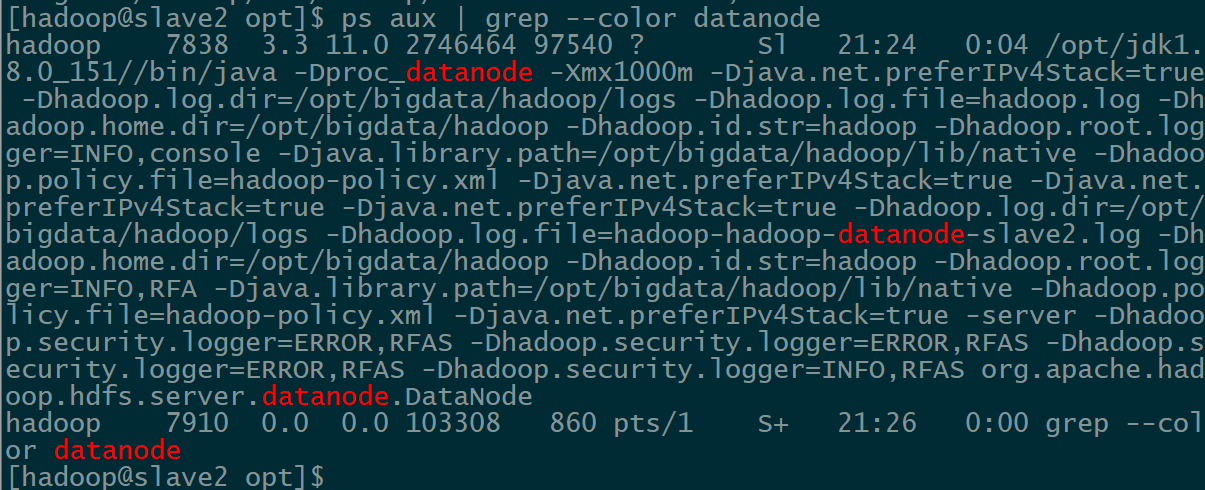
ps aux | grep --color namenode



#DataNode节点（slave1、slave2）上面查看进程

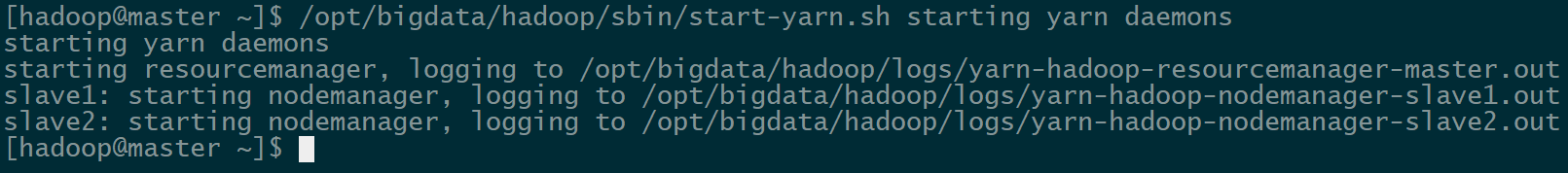
ps aux | grep --color datanode





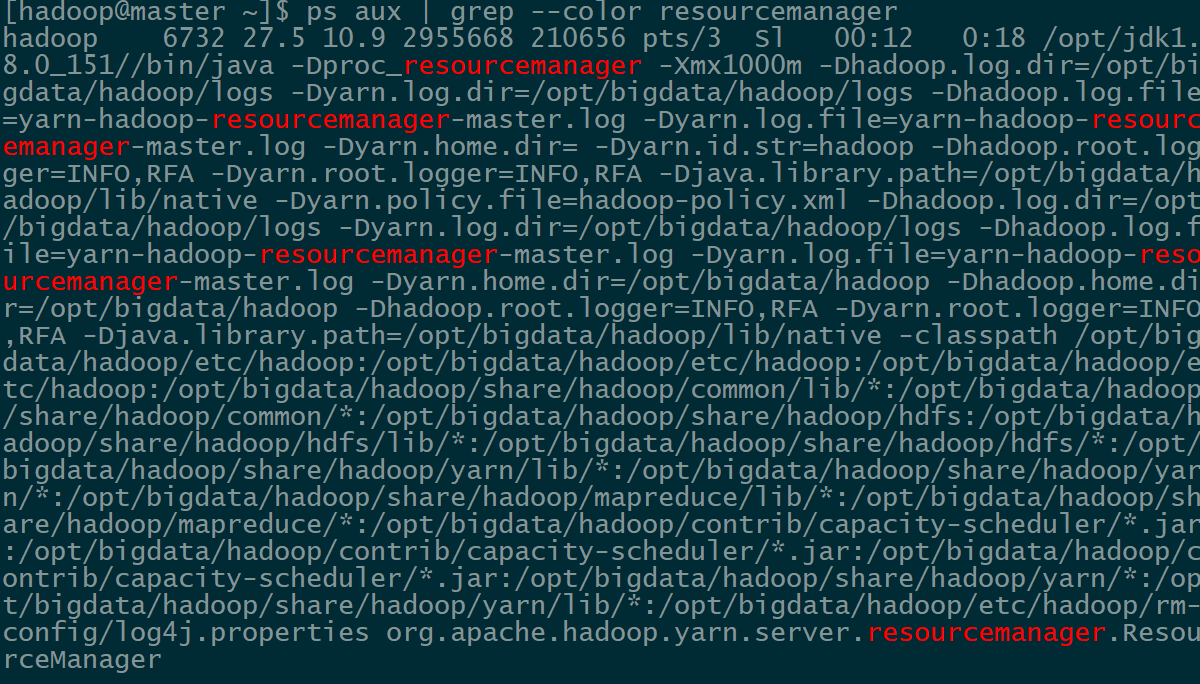
### 启动yarn分布式计算框架

[hadoop@master ~]$ start-yarn.sh starting yarn daemons



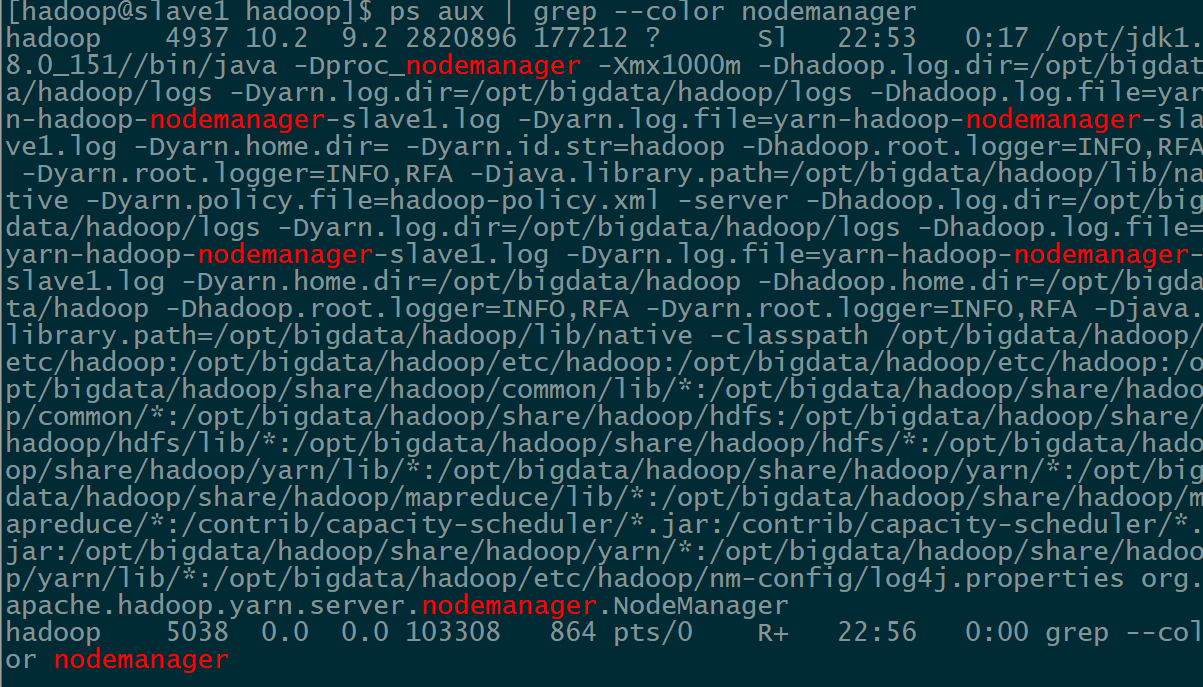
#NameNode节点（master）上查看进程

ps aux | grep --color resourcemanager



#DataNode节点（slave1、slave2）上查看进程

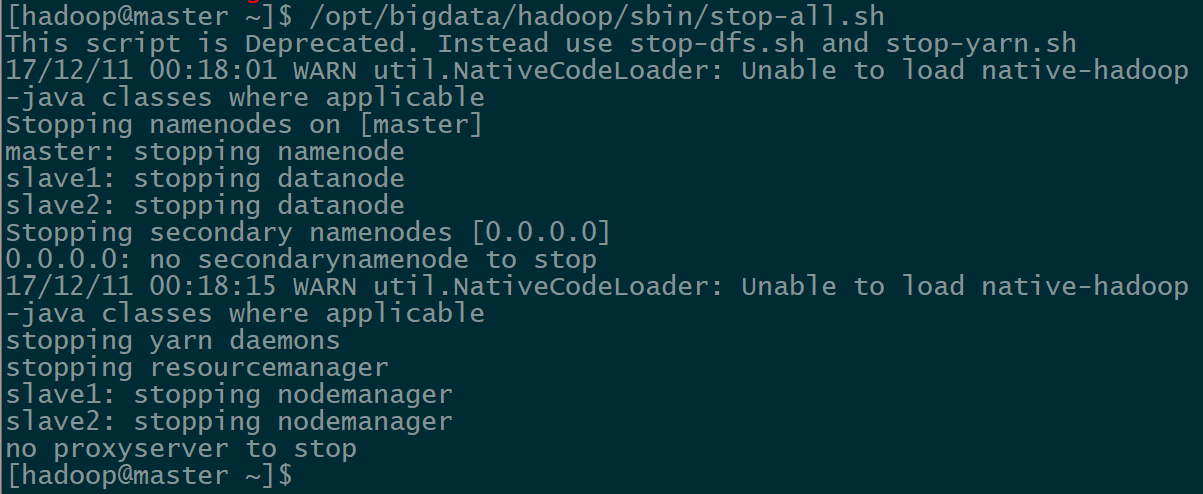
ps aux | grep --color nodemanager

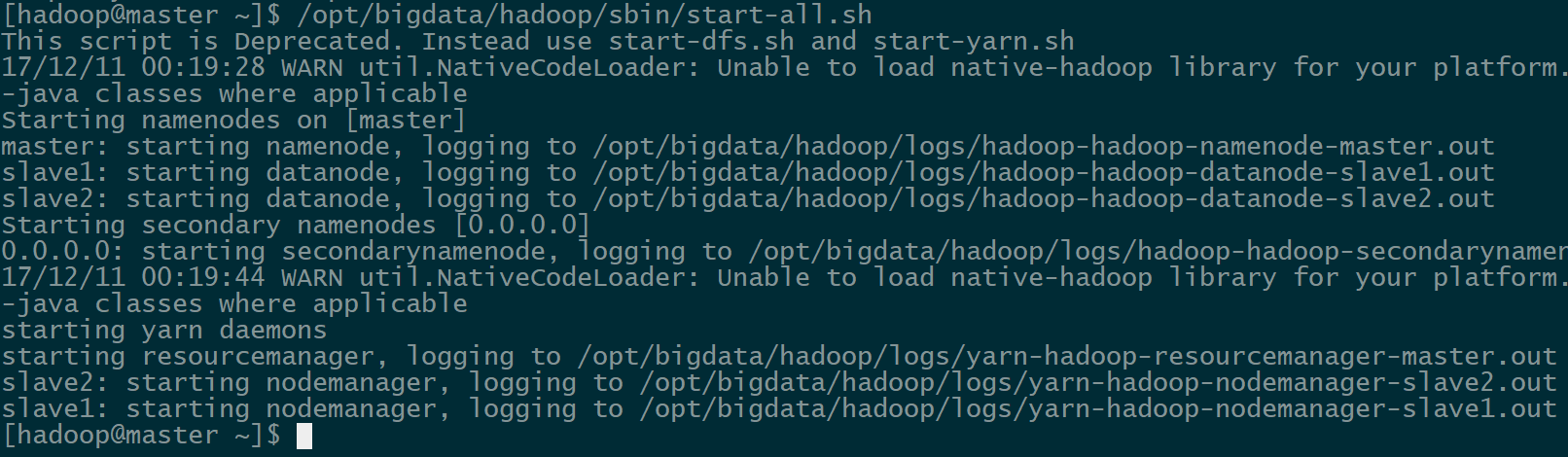


注：start-dfs.sh和start-yarn.sh这两个脚本可用start-all.sh代替

[hadoop@master ~]$ stop-all.sh

[hadoop@master ~]$ start-all.sh





### 启动jobhistory服务，查看mapreduce状态

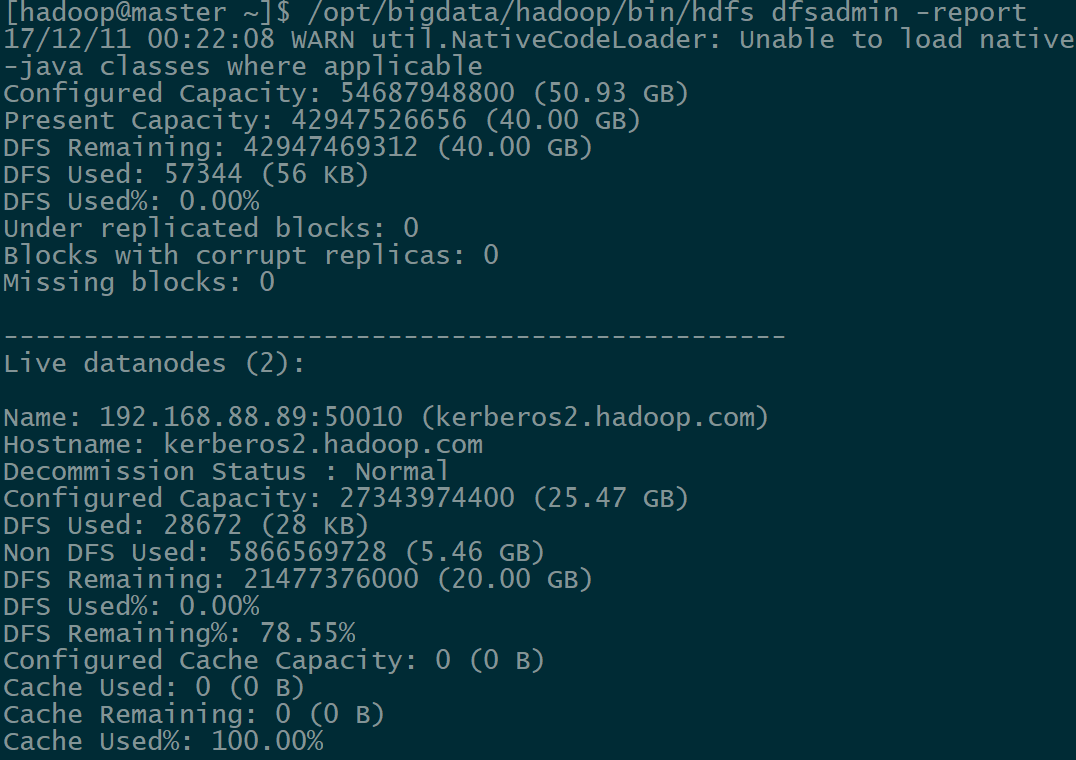
#在NameNode节点上

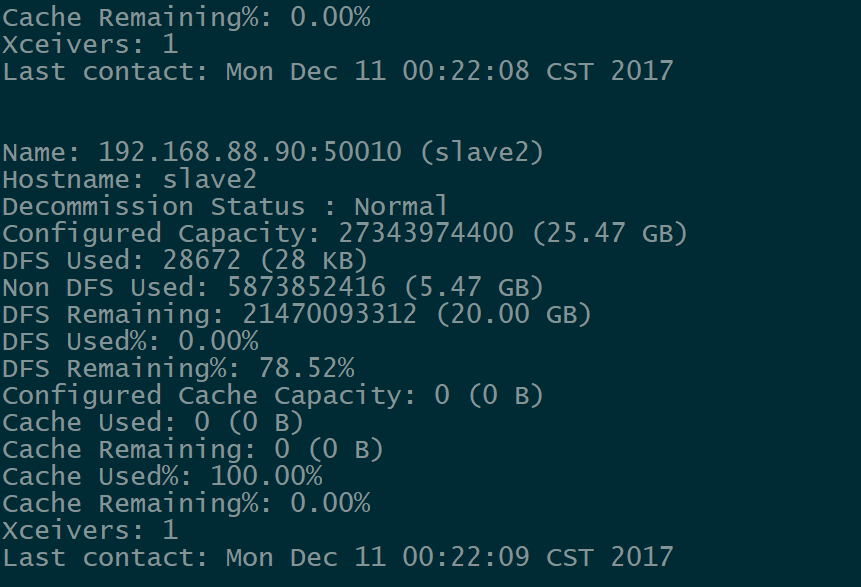
[hadoop@master ~]$ mr-jobhistory-daemon.sh start historyserver

starting historyserver, logging to /opt/bigdata/hadoop/logs/mapred-hadoop-historyserver-master.out

### 查看HDFS分布式文件系统状态

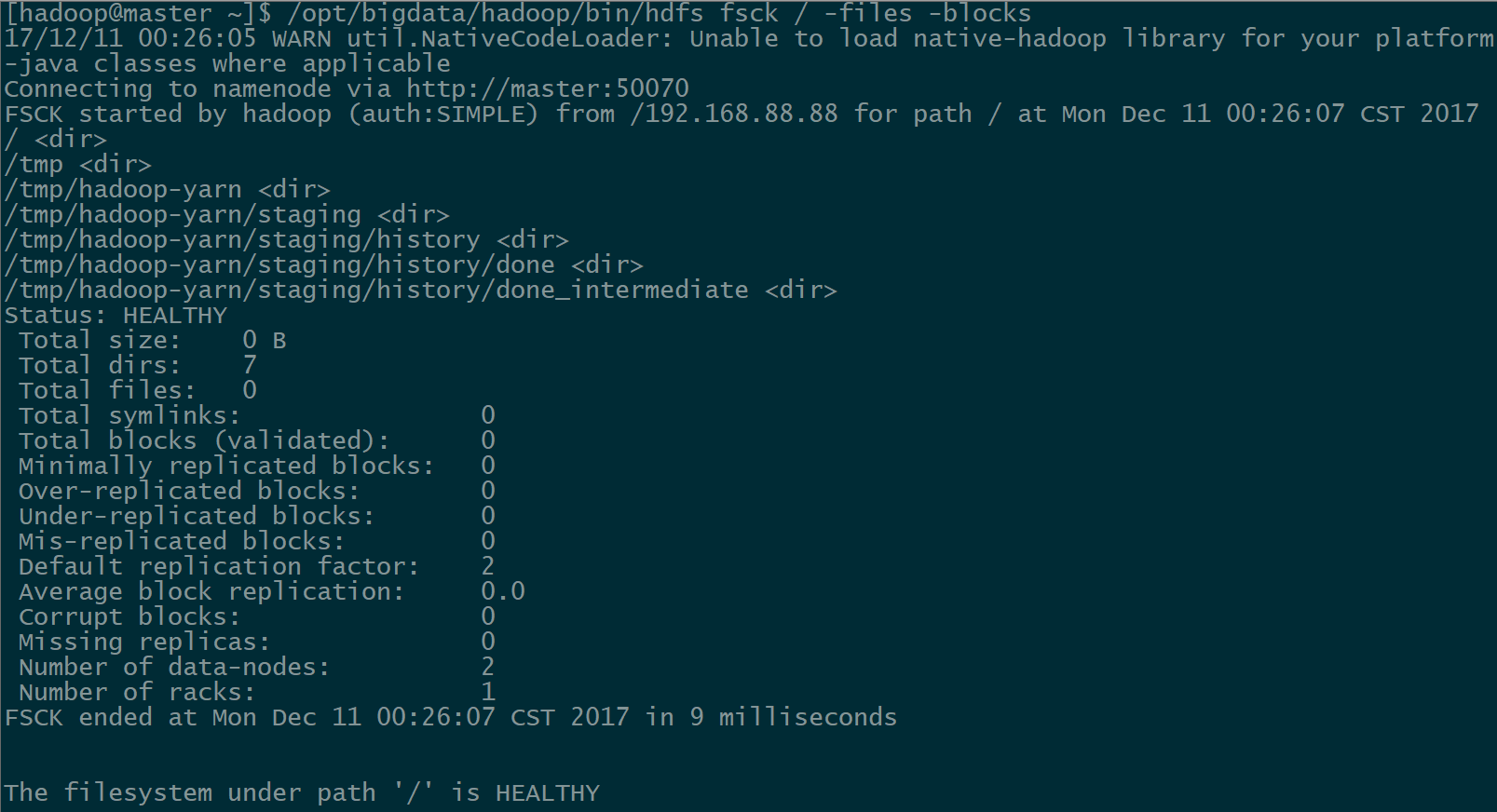
[hadoop@master ~]$ hdfs dfsadmin -report



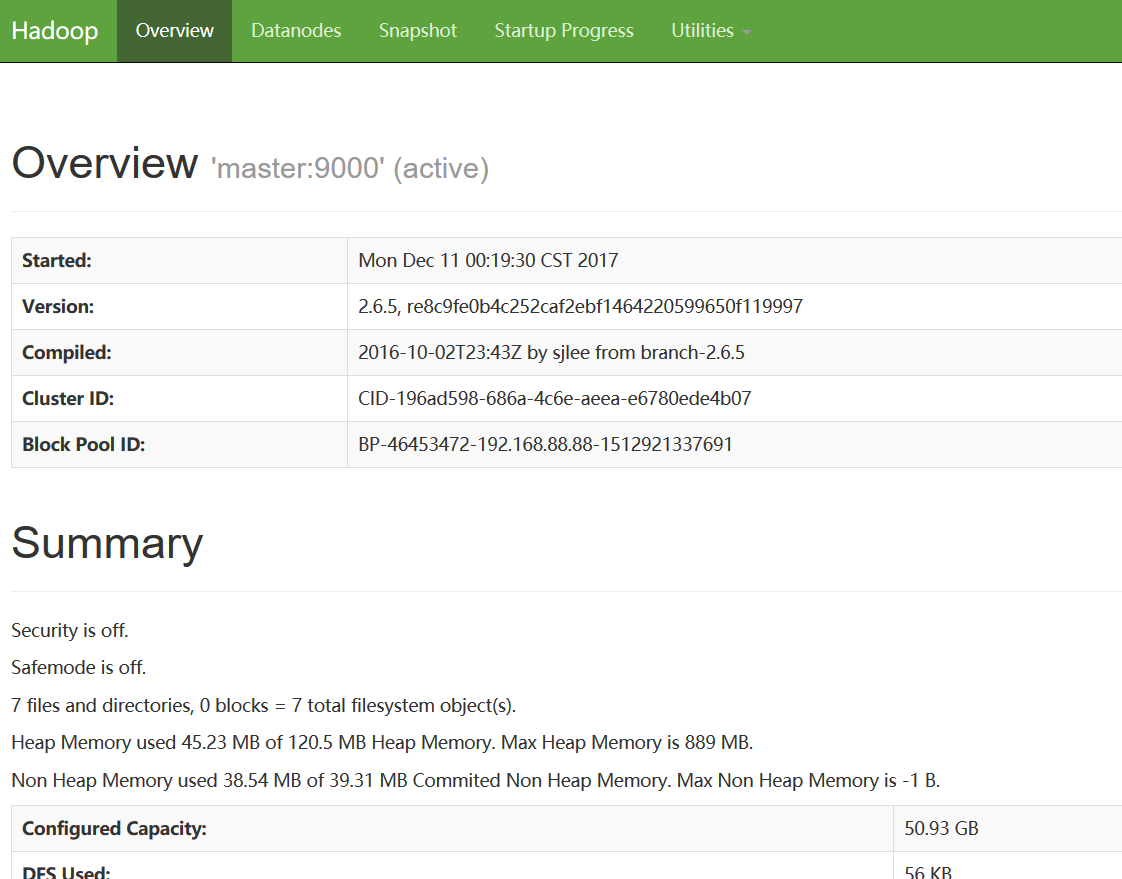
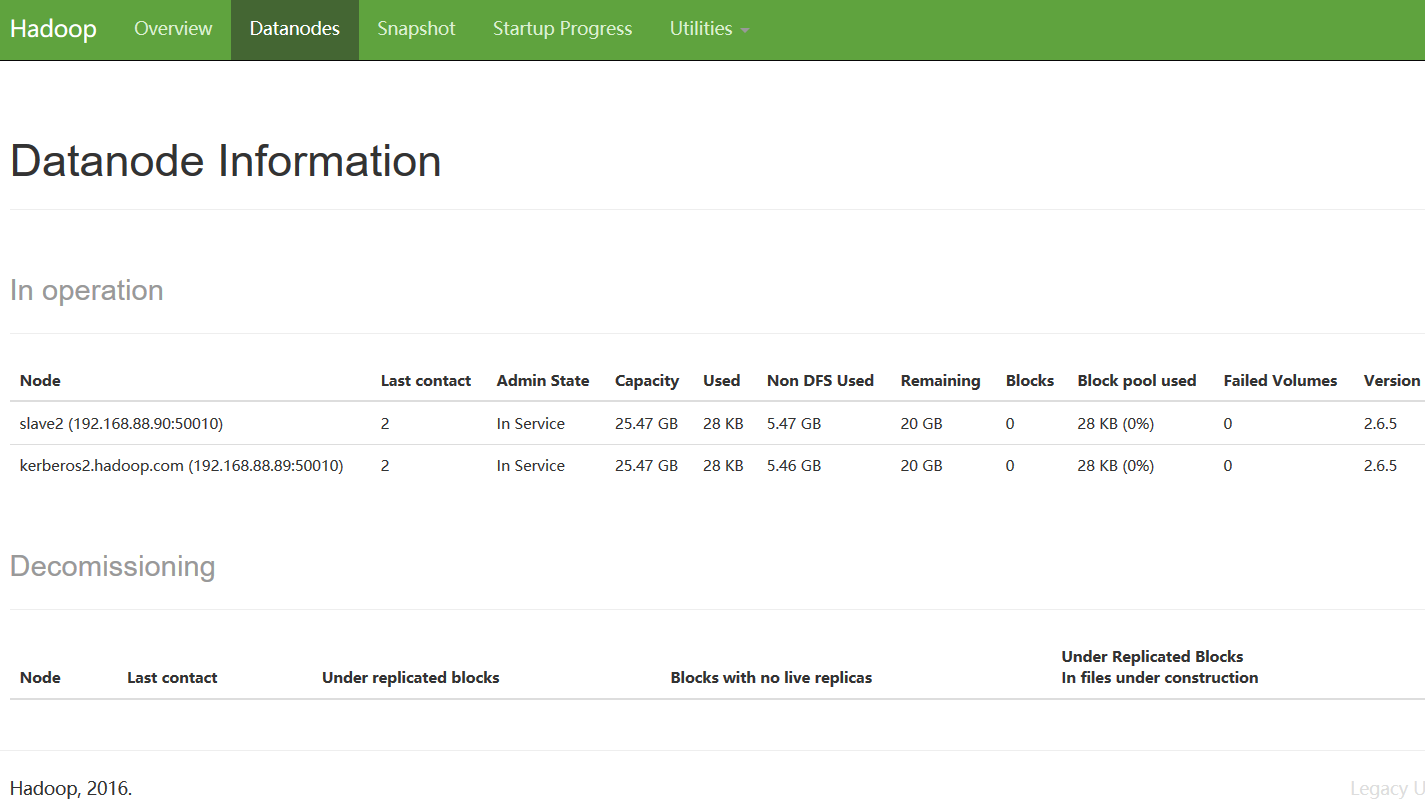


#查看文件块组成，一个文件由那些块组成

[hadoop@master ~]$ hdfs fsck / -files -blocks



### web页面查看hadoop集群状态

**查看HDFS状态**：http://192.168.88.88:50070/  
   
  
**查看Hadoop集群状态**：http://192.168.88.88:8088/  
