**搭建Hadoop分布式集群**

# 一、实验目的

* 理解Hadoop集群模式
* 掌握Hadoop分布式集群的配置
* 掌握服务器的使用以及Linux常见命令

# 二、实验内容

本实验基于CentOS 7.9.2009，使用Hadoop 3.3.3搭建分布式的集群环境。实验的包括基础环境的准备、jdk的安装部署、ssh免密码连接、Hadoop的安装和配置、使用Hadoop的基本操作、使用WebUI监控Hadoop运行等。通过实验，能够理解Hadoop的分布式架构的组成原理，掌握Hadoop环境的搭建过程，理解Hadoop配置文件的配置过程和每个属性的意义。为后续使用Hadoop进行程序设计和开发提供基本的运行环境，同时也可以更深刻理解Hadoop的组成和原理。

# 三、实验知识点

* Hadoop的安装和配置
* SSH免密码链接配置
* Jdk的安装和配置
* Linux的基本操作命令

# 四、实验环境

## 1. 软硬件环境

* Docker Desktop
* Centos 7.9.2009 64位
* Hadoop 3.3.3
* OpenJDK 1.8.0

## 2. 网络环境

集群包含3个节点：1个NameNode和2个DataNode，节点之间可以相互ping通。

# 五、实验步骤

## 1. 运行环境准备

### 1.1 验证服务器的hosts是否配置正确

确认hostname设置正确

在每台服务器上，执行如下命令确认host那么是否配置正确：

***cat /etc/hostname***

如果配置不正确，需要重新参考实验1的手册创建container。

### 1.2 安装必需的软件包

在每个container服务器上安装必需的软件包，执行如下命令：

***yum install -y java vim net-tools openssh openssh-clients openssh-server***

同时，执行如下命令开启sshd server。

***systemctl status sshd***

***systemctl start sshd***

***systemctl enable sshd***

三条命令分别为查看sshd服务的状态，如果已经启动就不再需要执行后面的命令。如果没有启动则需要首先启动，并设置为自动启动。

同时，还需要重新设置root的密码，以便后续使用，执行如下命令：

***passwd root***

### 1.3 配置SSH免密码连接

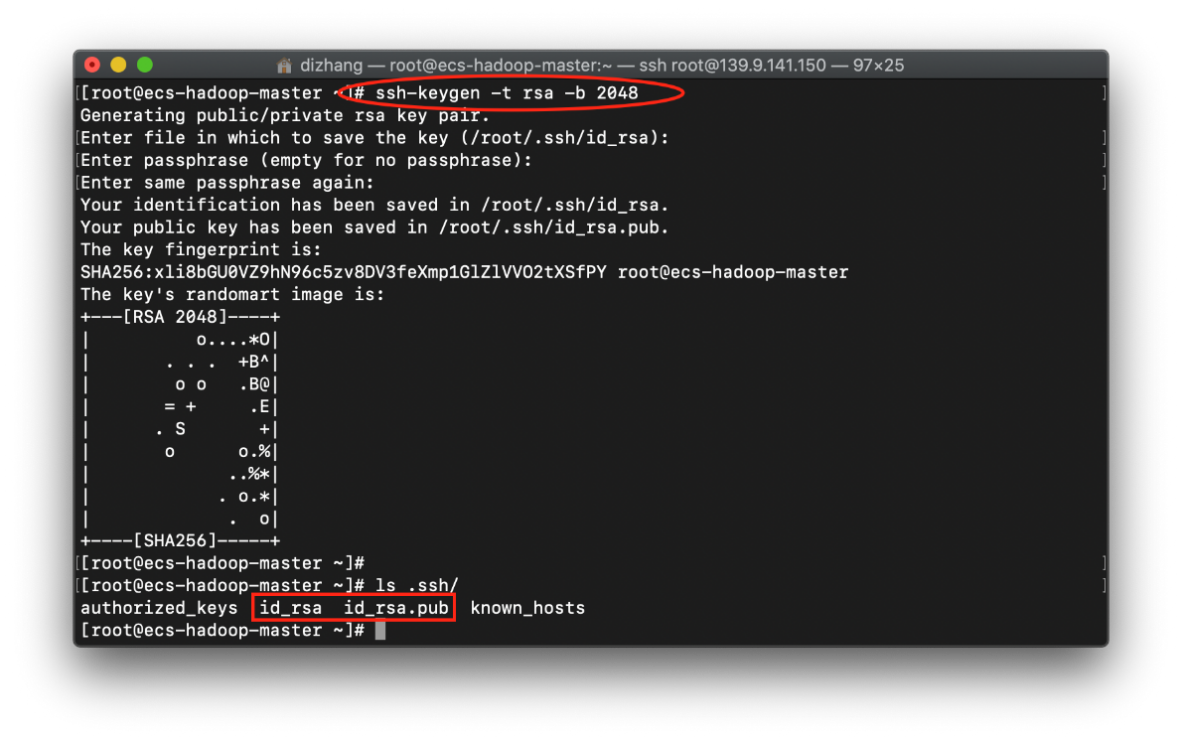
设置SSH免密码登录，是因为Hadoop需要通过SSH登录到各个节点进行操作，本手册使用操作的用户为root。如果使用的其他用户，请注意修改对应的用户名。

（1）生成rsa key

在每台服务器上，均执行如下命令：

***ssh-keygen -t rsa -b 2048***

有要求输入内容的时候，直接点击回车即可。该命令在.ssh目录下生成两个文件，分别为id\_rsa和id\_rsa.pub，其中id\_rsa为私有的key，id\_rsa.pub是公开的key。详细关于rsa的信息可以自行查阅。效果如下图所示。



（2）复制公开key到每个节点

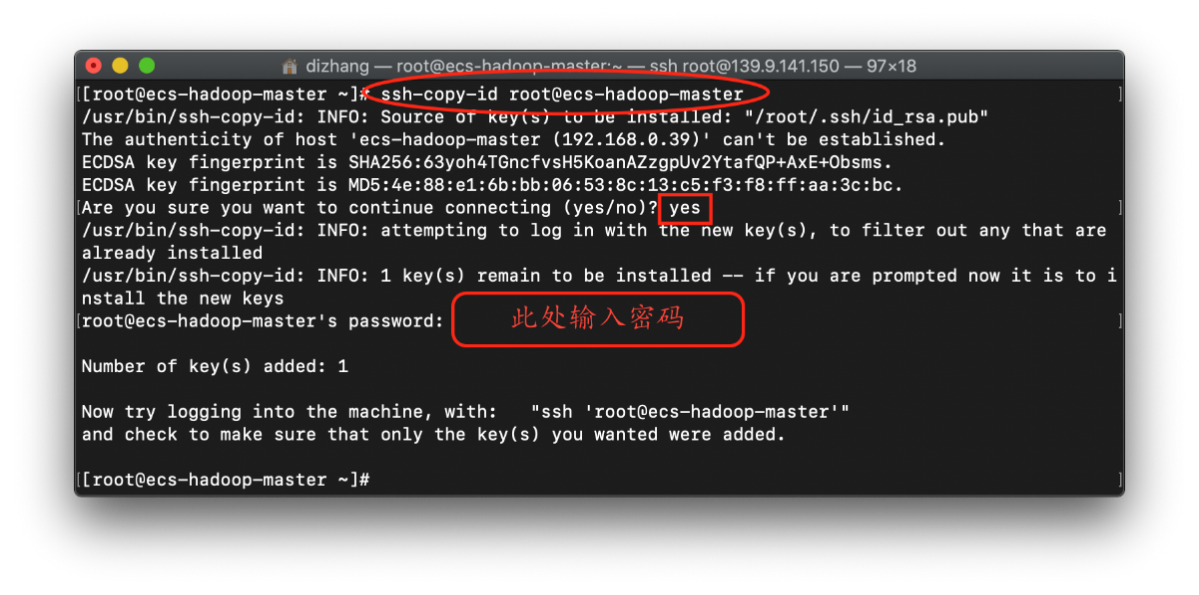
在每台服务器上，执行如下三条命令：

***ssh-copy-id root@hadoop-master***

***ssh-copy-id root@hadoop-slave1***

***ssh-copy-id root@hadoop-slave2***

如果提示“Are you sure you want to continue connecting”，输入yes，提示输入密码时，输入对应节点的密码即可。具体操作如下图所示。



（3）验证ssh免密码登录

在每台服务器上，输入如下三条命令，查看是否可以直接登录。如果前面配置没有问题的话，可以不用输入密码就可以直接登录hostname对应的节点。需要注意的是：如果登录成功，需要输入exit退出后再继续下一条测试。在一个节点测试完后，再到另外一个节点测试，直到三个节点均相互之间登录无问题。

***ssh root@hadoop-master***

***exit***

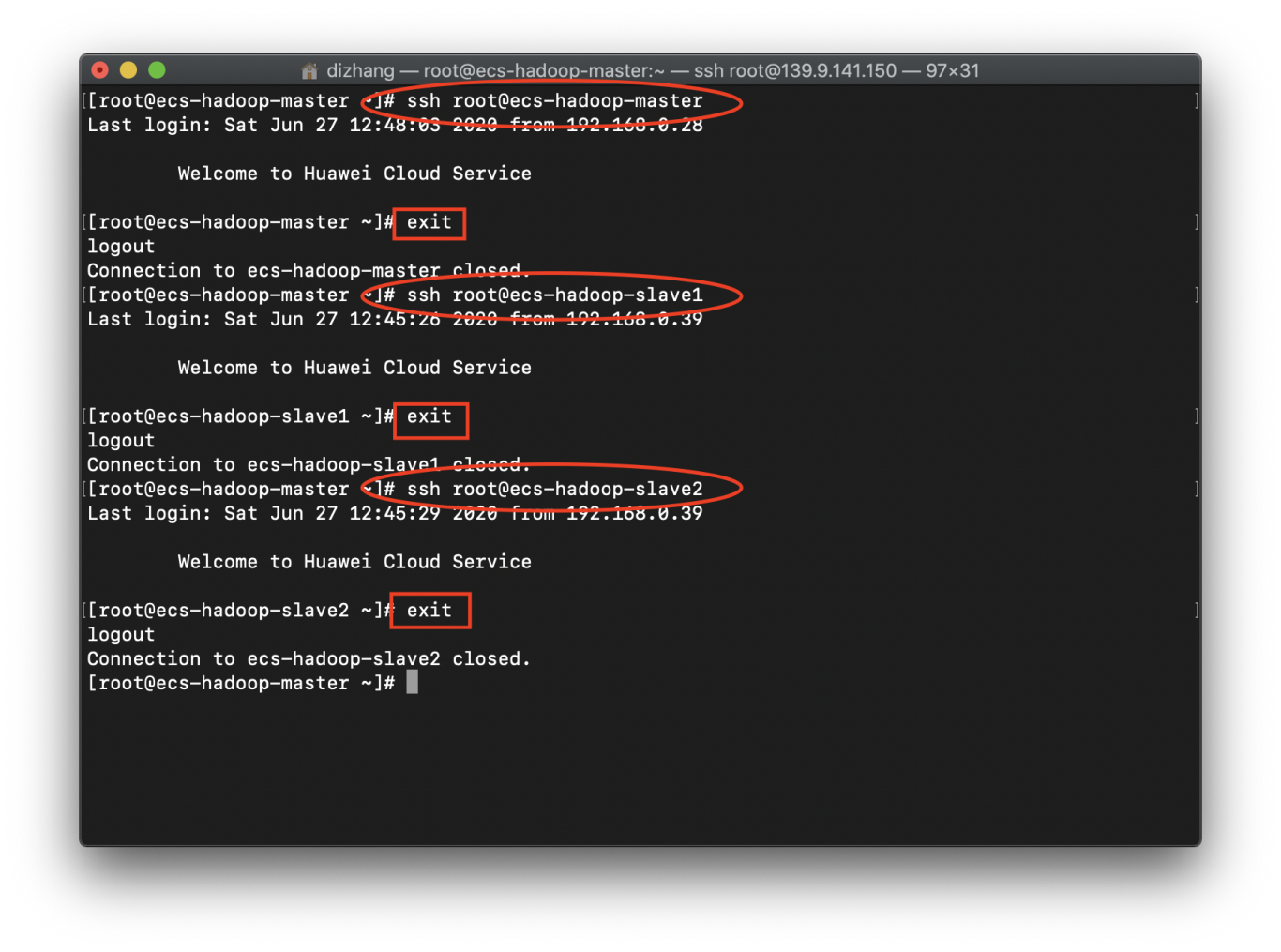
***ssh root@hadoop-slave1***

***exit***

***ssh root@hadoop-slave2***

***exit***

具体操作如下图所示。



### 1.4 安装并配置JDK

（1）安装openjdk-1.8

Hadoop需要JDK 1.8，可以通过下面的命令安装。

***yum install -y java-1.8.0-openjdk-devel***

（2）配置环境变量

在/etc/profile文件的最后添加如下内容：

执行命令：***vim /etc/profile***

***export JAVA\_HOME=/usr/lib/jvm//java-1.8.0***

***export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar***

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

然后，执行***source /etc/profile***命令。请注意，source执行的只对当前的登录临时有效，如果因为超时退出，配置的环境变量将失效。如果希望永久生效，请重启服务器。

## 2. 安装并配置Hadoop

本实验利用已经下载的hadoop-3.3.3.tar.gz进行安装配置。

（1）安装hadoop-3.3.3.tar.gz

首先，在home目录下创建hadoop目录。

***cd /home***

***mkdir hadoop***

然后，将hadoop-3.3.3.tar.gz复制到/home/hadoop目录下。执行如下命令。

tar -zxvf hadoop-3.3.3.tar.gz

cd hadoop-3.3.3

接着，在hadoop-3.3.3目录下创建tmp、dfs、dfs/data、dfs/name目录。

***mkdir tmp***

***mkdir -p dfs/data***

***mkdir -p dfs/name***

（2）配置Hadoop环境变量

编辑/etc/profile文件，添加如下内容：

***export HADOOP\_HOME=/home/hadoop/hadoop-3.3.3***

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

***export HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}***

***export HADOOP\_COMMON\_HOME=${HADOOP\_HOME}***

***export HADOOP\_HDFS\_HOME=${HADOOP\_HOME}***

***export YARN\_HOME=${HADOOP\_HOME}***

执行***source /etc/profile***命令，使得环境变量配置生效。

（3）配置hadoop-3.3.3

首先，进入到hadoop-3.3.3的目录。

***cd /home/hadoop/hadoop-3.3.3***

（a）配置core-site.xml文件，该文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下。在<configuration>标签下增加如下内容：

***<property>***

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

***<value>hdfs://hadoop-master:9000</value>***

***</property>***

***<property>***

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

***<value>/home/hadoop/hadoop-3.3.3/tmp</value>***

***</property>***

***<property>***

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

***<value>131702</value>***

***</property>***

（b）配置hdfs-site.xml，该文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下。在<configuration>标签下增加如下内容：

***<property>***

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

***<value>/home/hadoop/hadoop-3.3.3/dfs/name</value>***

***</property>***

***<property>***

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

***<value>/home/hadoop/hadoop-3.3.3/dfs/data</value>***

***</property>***

***<property>***

***<name>dfs.replication</name>***

***<value>2</value>***

***</property>***

***<property>***

***<name>dfs.permissions.enable</name>***

***<value>false</value>***

***</property>***

***<property>***

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

***<value>true</value>***

***</property>***

（c）配置mapred-site.xml文件，该文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下。在<configuration>标签下增加如下内容：

***<property>***

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

***<value>yarn</value>***

***</property>***

***<property>***

***<name>yarn.app.mapreduce.am.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop-3.3.3</value>***

***</property>***

***<property>***

***<name>mapreduce.map.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop-3.3.3</value>***

***</property>***

***<property>***

***<name>mapreduce.reduce.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop-3.3.3</value>***

***</property>***

（d）配置yarn-site.xml文件，该文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下。在<configuration>标签下增加如下内容：

（e）配置hadoop-env.sh文件，该文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下。在文件中增加JAVA\_H***<property>***

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

***<value>3192</value>***

***</property>***

***<property>***

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

***<value>5</value>***

***</property>***

***<property>***

***<name>yarn.scheduler.minimum-allocation-mb</name>***

***<value>512</value>***

***</property>***

***<property>***

***<name>yarn.scheduler.maximum-allocation-mb</name>***

***<value>1536</value>***

***</property>***

***<property>***

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

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

***</property>***

***<property>***

***<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>***

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

***</property>***

***<property>***

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

***<value>hadoop-master</value>***

***</property>***

OME变量，如果不设置，无法正常启动集群。

***vim hadoop-env.sh***

在如下的位置，增加JAVA\_HOME配置。

***# export JAVA\_HOME=***

***export JAVA\_HOME=/usr/lib/jvm/java/***

（f）配置workers和masters文件，其中workers文件位于/home/hadoop/hadoop-3.3.3/etc/hadoop/目录下，masters文件在该目录下自行创建。

输入命令***vim workers***，删除问价中默认的localhost，添加如下两个slave节点的IP地址，注意替换成自己服务器的IP地址。

***172.18.0.3***

***172.18.0.4***

使用命令vim masters，增加master节点的IP地址，注意替换成自己服务器的IP地址。

***172.18.0.2***

（g）修改启动命令，以支持root启动hadoop

启动命令在/home/hadoop/hadoop-3.3.3/sbin下。执行如下命令，改变当前目录：

***cd /home/hadoop/hadoop-3.3.3/sbin***

然后，利用vim工具修改start-dfs.sh、stop-dfs.sh、start-yarn.sh和stop-yarn.sh四个文件，注意修改的内容放到对应脚本文件内容#!/usr/bin/env bash的下一行。

其中，在start-dfs.sh、stop-dfs.sh文件中分别增加如下内容：

***HDFS\_DATANODE\_USER=root***

***HDFS\_DATANODE\_SECURE\_USER=hdfs***

***HDFS\_NAMENODE\_USER=root***

***HDFS\_SECONDARYNAMENODE\_USER=root***

接着，修改start-yarn.sh和stop-yarn.sh，分别增加如下内容：

***YARN\_RESOURCEMANAGER\_USER=root***

***HDFS\_DATANODE\_SECURE\_USER=yarn***

***YARN\_NODEMANAGER\_USER=root***

（4）通过scp将配置好的Hadoop复制到其他节点对应目录

执行如下命令：

***scp -r /home/hadoop/ root@hadoop-slave1:/home***

***scp -r /home/hadoop/ root@hadoop-slave2:/home***

## 3. 启动Hadoop集群并测试

## 3.1 启动和停止hadoop集群

（1）格式化namenode

首次启动hadoop之前需要进行格式化。因此，第一次启动时，先执行如下命令：

***hdfs namenode -format***

（2）启动hadoop集群

执行***start-all.sh***命令可以启动全部hadoop的所有守护进程，也可以分开启动各个进程。分别执行***start-dfs.sh***和***start-yarn.sh***。

（3）查看master节点和datanode节点的守护进程

通过***jps***命令可以查看hadoop节点上启动的守护进程。如果成功启动的话，master节点有NameNode、SecondaryNameNode、ResourceManager、Jps共4个守护进程。Datanode节点上有DataNode、NodeManager、Jps共3个守护进程。

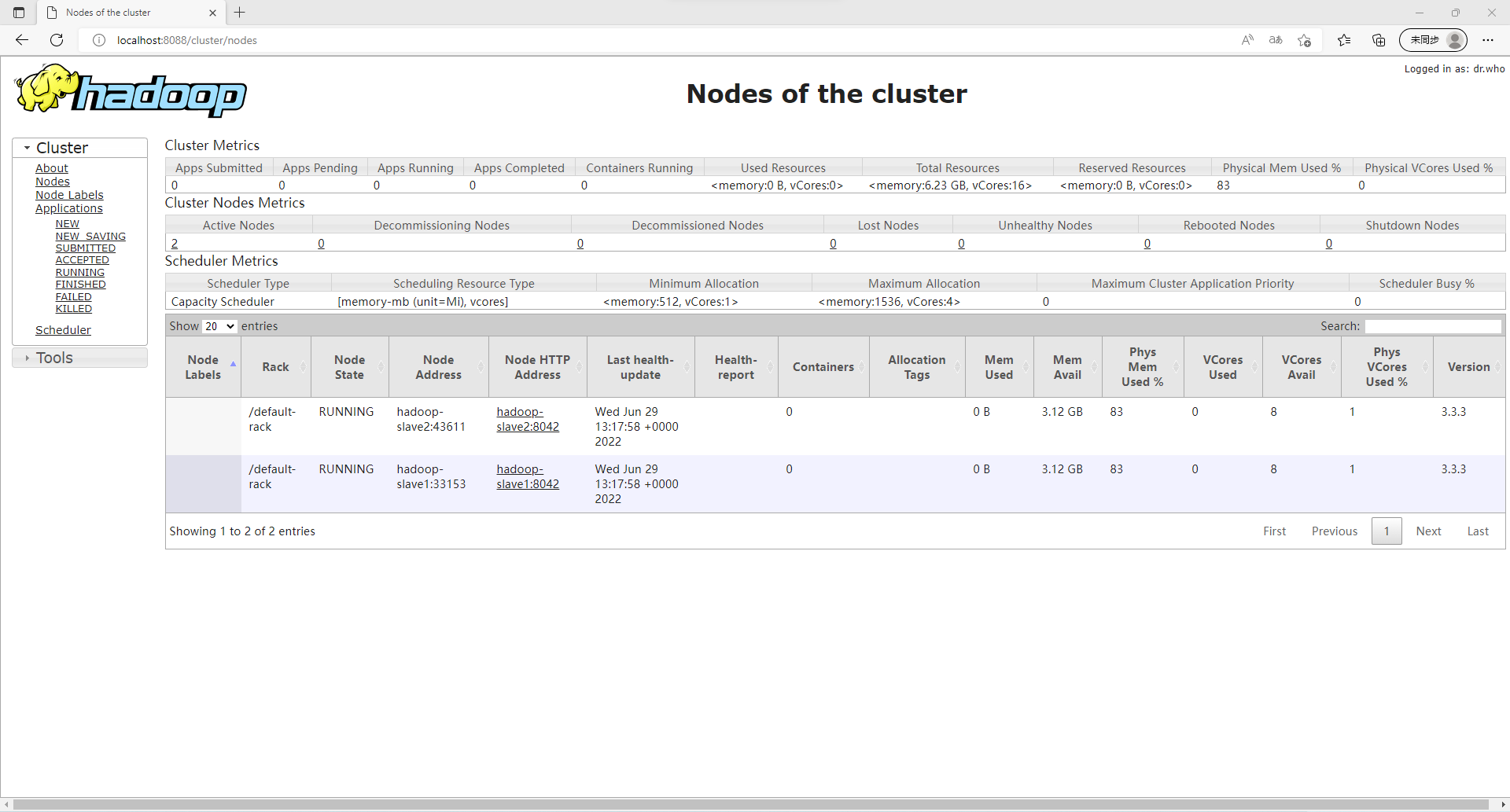
（4）停止hadoop集群

执行***stop-all.sh***命令可以停止hadoop的所有守护进程，也可以分别停止各个进程。分别执行***stop-yarn.sh***和***stop-dfs.sh***。

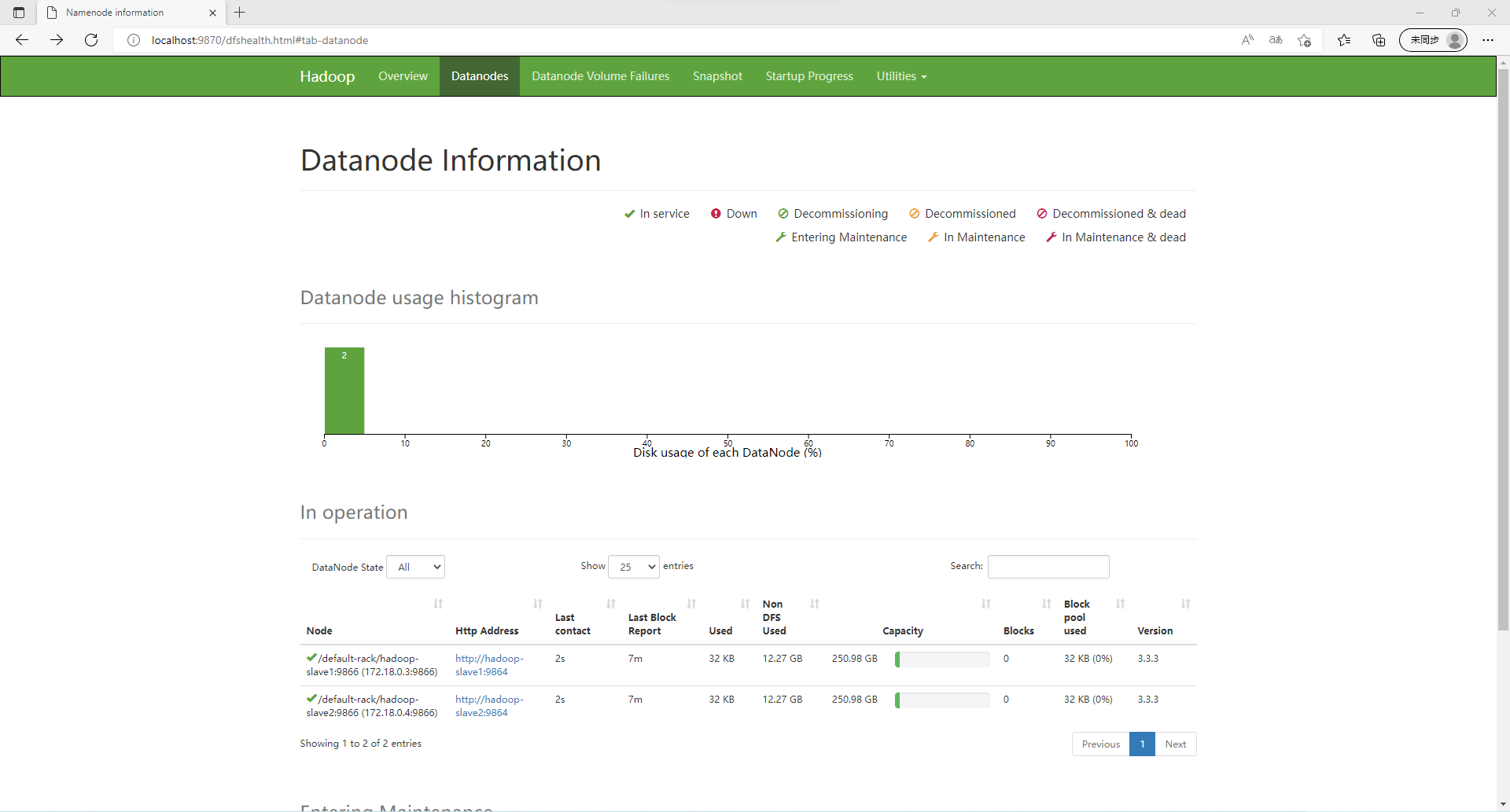
### 3.2 查看Hadoop集群是否启动成功

可以通过WebUI来查看Hadoop集群是否启动成功。

首先，在浏览器中http://localhost:8088/。如果配置成功则能够看到datanode节点。如下图所示。



然后，在浏览器中输入http://localhost:9870/。如果配置成功，在此页面可以看到namenode的状态，以及datanode的状态。点击Datanodes也可以看到配置的所有datanode的状态。具体效果分别如下图所示。



至此，Hadoop的安装配置完成。但这只是大数据应用的开始，之后还要根据自己的需求，编写程序调用Hadoop的接口，发挥HDFS和MapReduce的作用。

## 4. 可能遇到的问题及解决方案

Hadoop启动的过程中，会将在hadoop目录下的logs文件夹中记录启动日志，可以通过查看日志的内容来定位问题，根据日志的具体内容查找问题的所在和可能的解决方案。

（1）Datanode无法启动

首先，要执行stop-all.sh停止所有服务。然后将所有slave节点上的tmp、logs、文件夹删除，并重新创建这两个文件夹。之后，重新格式化：hdfs namenode -format。最后，再调用start-all.sh启动。

（2）遇到问题的基本从以下几个方面来解决：

* 检查各个XML文件是否配置正确；
* Java环境变量是否配置正确；
* SSH是否可以无密码连接。