Hadoop(CDH)分布式环境搭建(简单易懂,绝对有效)

大数据技术与架构 5月6日

本文是由alice菌发表在：https://blog.csdn.net/weixin_44318830/article/details/102846055

这篇博客,小菌分享的是大数据集群的安装部署,超级有效,希望能够帮助到大家!在正式部署之前,我们需要做一些准备工作。

准备好三台虚拟机,ip分别为192.168.100.100,192.168.100.110,192.168.100.120。对应的主机

服务器 IP	192.168.100.100	192.168.52.110	192.168.100.120
HDFS	NameNode		
	SecondaryNameNode		
	DataNode	DataNode	DataNode
YARN	ResourceManager		
	NodeManager	NodeManager	NodeManager
MapReduce	JobHistoryServer		


```
三台机器准备工作:
1.关闭防火墙    service iptables stop
              chkconfig iptables stop
2.关闭selinux
3.修改主机名
4.ssh无密码拷贝数据
特别说明(在主节点无密码访问到从节点)
ssh-keygen
ssh-copy-id   192.168.100.100
ssh-copy-id   192.168.100.110
ssh-copy-id   192.168.100.120
5.设置主机名和IP对应
vi  /etc/hosts
6.jdk1.8安装
```

一.上传压缩包并解压

二.查看Hadoop支持的压缩方式以及本地库

三.修改配置文件
1.修改core-site.xml
2.修改hdfs-site.xml
3.修改Hadoop-env.sh
4.修改mapred-site.xml
5.修改yarn-site.xml
6.修改slaves文件
四.创建文件存放目录
五:安装包的分发
六:配置Hadoop的环境变量
七:集群启动
1.单个节点逐一启动
2.脚本一键启动HDFS、Yarn
3.脚本一键启动所有
八:浏览器查看启动页面
九:验证集群是否可用
十:HDFS初体验

一.上传压缩包并解压

创建两个文件夹

```
mkdir -p /export/softwares 存放软件压缩包
mkdir -p /export/servers 存放压缩后的文件
```

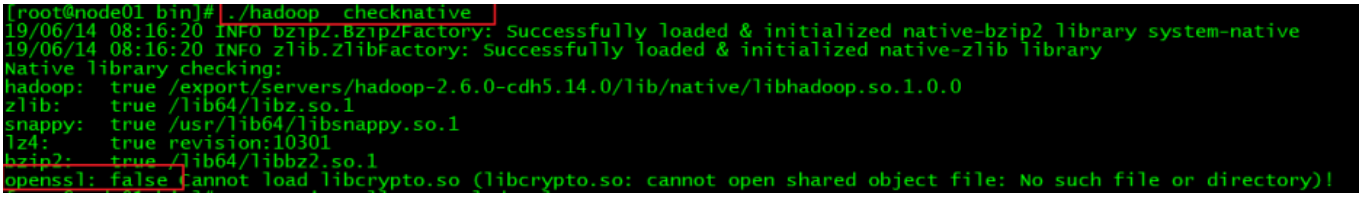
将我们的hadoop安装包上传到第一台服务器的/export/software并解压至/export/servers。
第一台机器执行以下命令:

```
cd /export/softwares/
mv hadoop-2.6.0-cdh5.14.0-自己编译后的版本.tar.gz hadoop-2.6.0-cdh5.14.0.ta
tar -zxvf hadoop-2.6.0-cdh5.14.0.tar.gz -C    ../servers/
```

二.查看Hadoop支持的压缩方式以及本地库

第一台机器执行以下命令:

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0
bin/hadoop checknative
```



如果出现openssl为false，那么所有机器在线安装openssl即可，执行以下命令，虚拟机联网之后就可以在线进行安装了。

```
yum -y install openssl-devel
```

三.修改配置文件

这一步对于CDH分布式环境的搭建非常重要,大家在修改配置文件的时候一定要谨慎！

1.修改core-site.xml

第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
vim core-site.xml
```

将以下内容添至xml文件指定位置处

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://node01:8020</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/tempDatas
  </property>
  <!--  缓冲区大小，实际工作中根据服务器性能动态调整 -->
  <property>
    <name>io.file.buffer.size</name>
    <value>4096</value>
  </property>
```

```
15
16    <!--  开启hdfs的垃圾桶机制，删除掉的数据可以从垃圾桶中回收，单位分钟 -->
17    <property>
18      <name>fs.trash.interval</name>
19      <value>10080</value>
20    </property>
21  </configuration>
```

注意事项:

2.修改hdfs-site.xml

第一台机器执行以下命令

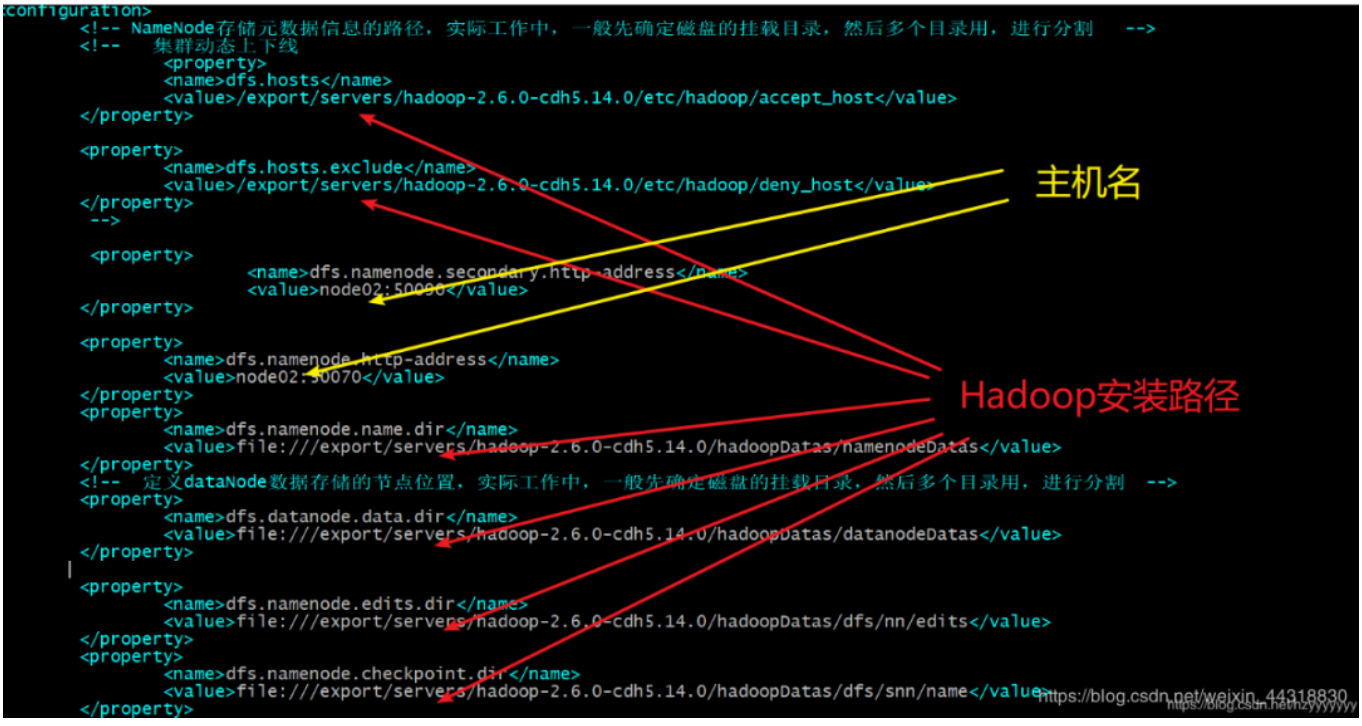
```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
vim hdfs-site.xml
```

将以下内容添至xml文件中

```
<configuration>
  <!-- NameNode存储元数据信息的路径，实际工作中，一般先确定磁盘的挂载目录，然后多个
  <!--   集群动态上下线
  <property>
    <name>dfs.hosts</name>
    <value>/export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/accept_hc
  </property>

  <property>
    <name>dfs.hosts.exclude</name>
    <value>/export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/deny_host
  </property>
   -->

   <property>
      <name>dfs.namenode.secondary.http-address</name>
      <value>node01:50090</value>
  </property>

  <property>
    <name>dfs.namenode.http-address</name>
    <value>node01:50070</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/r
  </property>
  <!--  定义dataNode数据存储的节点位置，实际工作中，一般先确定磁盘的挂载目录，然后
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/d
  </property>

  <property>
    <name>dfs.namenode.edits.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/d
  </property>
  <property>
    <name>dfs.namenode.checkpoint.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/d
```

```
  </property>
  <property>
    <name>dfs.namenode.checkpoint.edits.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/c
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
<property>
    <name>dfs.blocksize</name>
    <value>134217728</value>
  </property>
</configuration>
```

注意事项:



3.修改Hadoop-env.sh

第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
```

```
vim hadoop-env.sh
```

进入文件后在export JAVA_HOME={JAVA_HOME} 这行代码中修改成jdk的安装路径

```
export JAVA_HOME=/export/servers/jdk1.8.0_141
```

4.修改mapred-site.xml

第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
vim mapred-site.xml
```

将以下内容添至xml文件指定位置处中

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

  <property>
    <name>mapreduce.job.ubertask.enable</name>
    <value>true</value>
  </property>

  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>node01:10020</value>
  </property>

  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>node01:19888</value>
  </property>
</configuration>
```

注意事项:

5.修改yarn-site.xml

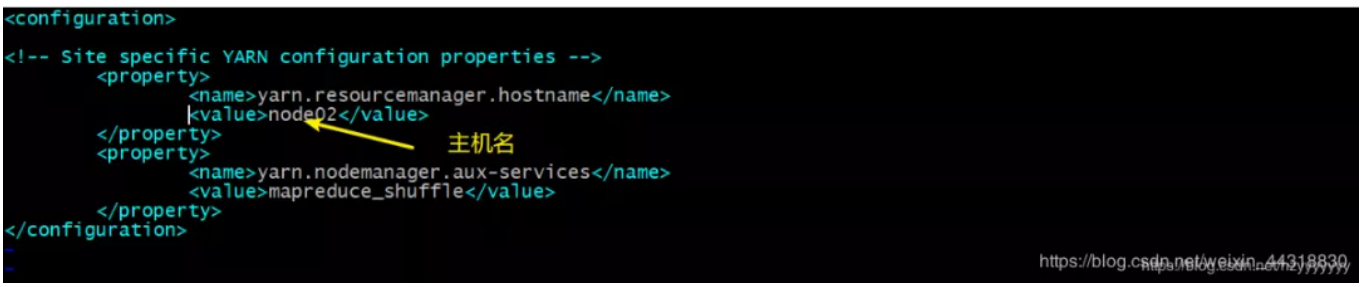
第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
vim yarn-site.xml
```

将以下内容添至xml文件指定位置处中

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>node01</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>

```

注意事项:



6.修改slaves文件

第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop
vim slaves
```

进入文件后写上你集群对应的主机名:

```
node01
node02
node03
```

四.创建文件存放目录

第一台机器执行以下命令，用来在node01机器上创建以下目录

```
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/tempDatas
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/namenodeDatas
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/datanodeDatas
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/edits
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/snn/name
mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/snn/ed
```

五:安装包的分发

第一台机器执行以下命令

```
cd /export/servers/
```

```
scp -r hadoop-2.6.0-cdh5.14.0/ node02:$PWD
scp -r hadoop-2.6.0-cdh5.14.0/ node03:$PWD
```

六:配置Hadoop的环境变量

三台机器都要进行**配置Hadoop的环境变量**
三台机器执行以下命令

```
vim  /etc/profile.d/hadoop.sh
```

```
export HADOOP_HOME=/export/servers/hadoop-2.6.0-cdh5.14.0
export PATH=:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

配置完成之后生效

```
source /etc/profile
```

七:集群启动

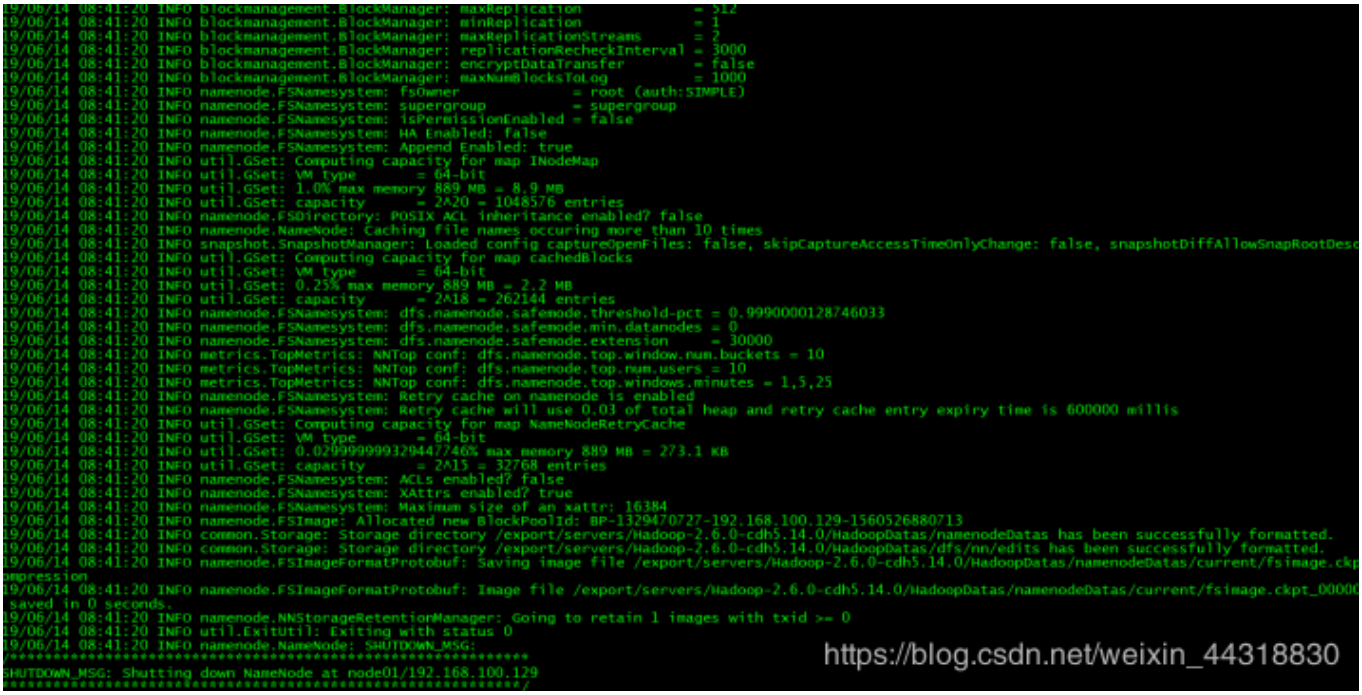
要启动 Hadoop 集群，需要启动 HDFS 和 YARN 两个集群。

注意：首次启动HDFS时，必须对其进行格式化操作。本质上是一些清理和准备工作，因为此时的 HDFS 在物理上还是不存在的。

bin/hdfs namenode -format

提示:

不要轻易格式化集群,格式化后集群的数据丢失且无法恢复



启动方式分为三种:

1.单个节点逐一启动

```
在主节点上使用以下命令启动 HDFS NameNode：Hadoop-daemon.sh start namenode
在每个从节点上使用以下命令启动 HDFS DataNode：Hadoop-daemon.sh start datanode
在主节点上使用以下命令启动 YARN ResourceManager：yarn-daemon.sh  start resourc
在每个从节点上使用以下命令启动 YARN nodemanager：yarn-daemon.sh start nodemana
```

以上脚本位于$HADOOP_PREFIX/sbin/目录下。如果想要停止某个节点上某个角色，只需要把命令中的start 改为stop 即可。

2.脚本一键启动HDFS、Yarn

如果配置了 etc/Hadoop/slaves 和 ssh 免密登录，则可以使用程序脚本启动所有Hadoop 两个集群的相关进程，在主节点所设定的机器上执行。

启动集群

node01节点上执行以下命令

第一台机器执行以下命令

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/
sbin/start-dfs.sh    #  开启HDFS
sbin/start-yarn.sh   #  开启Yarn
```

停止集群:

没事不要去停止集群

```
sbin/stop-dfs.sh
sbin/stop-yarn.sh
```

3.脚本一键启动所有

```
cd /export/servers/hadoop-2.6.0-cdh5.14.0/
```

一键启动集群

```
sbin/start-all.sh
```

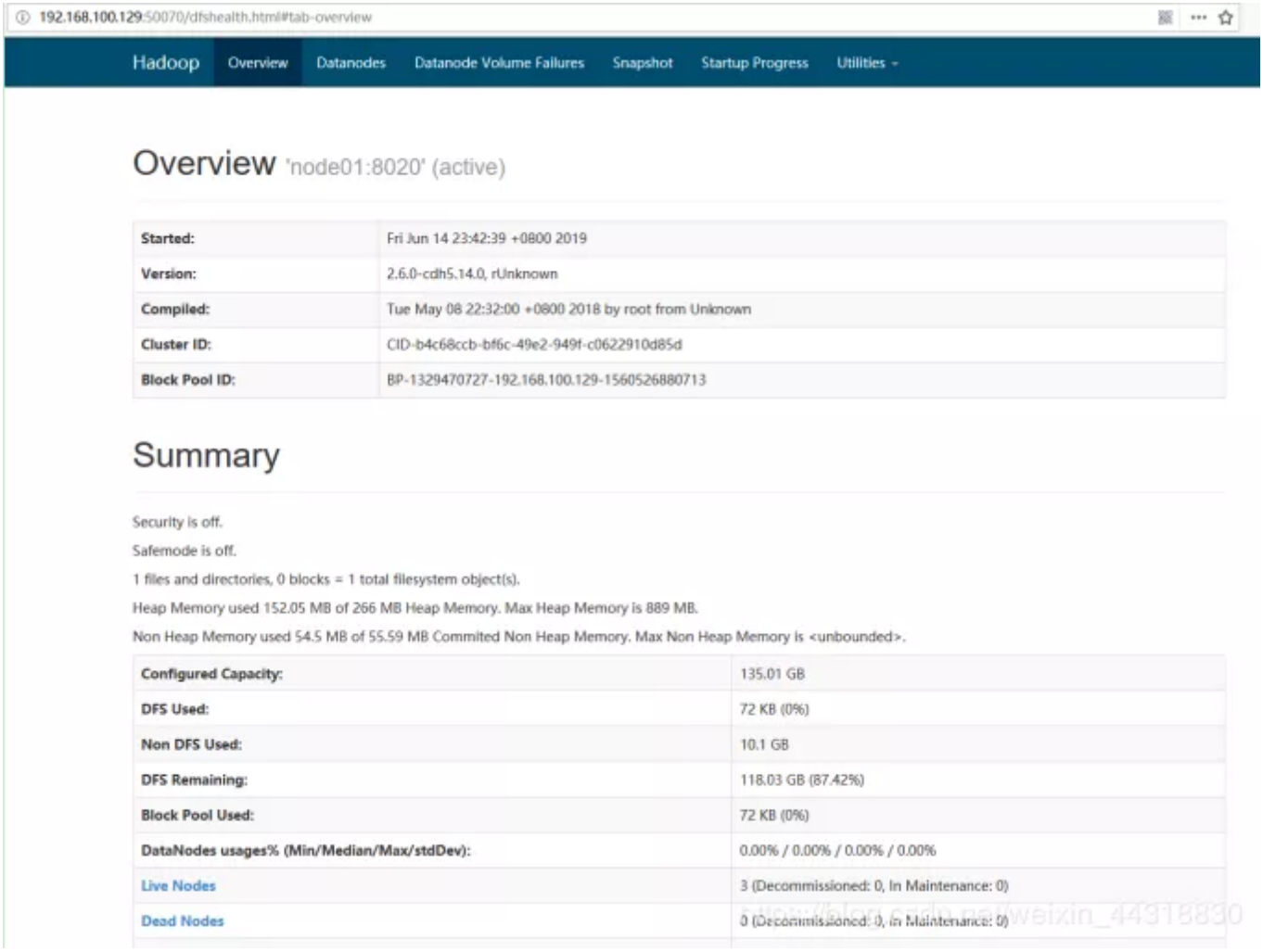
一键关闭集群

```
sbin/stop-all.sh
```

八:浏览器查看启动页面

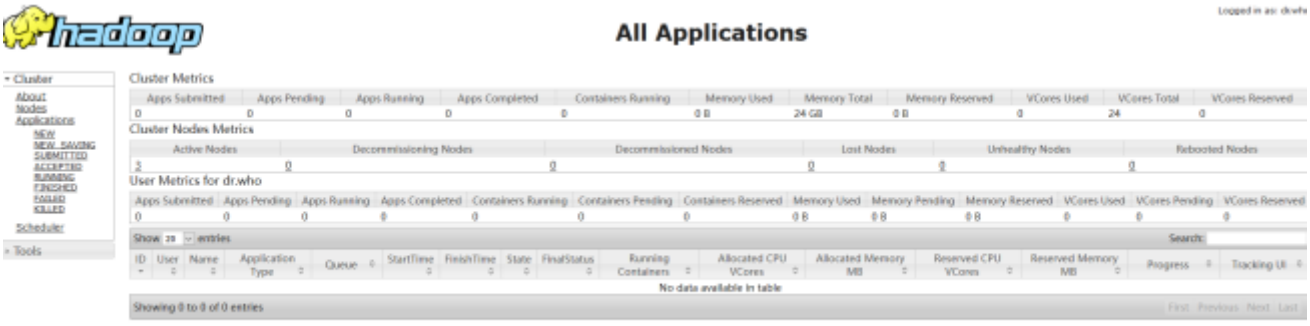
hdfs集群访问地址：

```
# 填写node01的ip
http://192.168.100.100:50070/dfshealth.html#tab-overview
```



yarn集群访问地址：

```
# 填写node01的ip
http://192.168.52.100:8088/cluster
```



如果看见以上画面，配置就全部成功了。

另外,我们通过命令jps也可以验证:

第一台主机:

```
[root@node01 softwares]# jps
30272 NameNode
31586 Jps
30665 ResourceManager
30762 NodeManager
30524 SecondaryNameNode
30399 DataNode
[root@node01 softwares]#
```


其他机器:

```
[root@node02 ~]# jps
28096 DataNode
28600 Jps
28190 NodeManager
[root@node02 ~]#
```

如果截图内容与上图不符，可能是在前面的步骤中哪里出错了,还需要重新检查一遍!

九:验证集群是否可用

常见的方式有以下几种:

```
1.jps用于验证集群服务的启动情况
2.namenode所在节点的IP+50070端口查看HDFS的web界面是否可用
3.在HSFS系统中创建一个文件夹或文件,若能创建表示集群可以正常使用!
   需要注意的是:
   HDFS不支持目录或文件夹的切换,所有路径必须写成绝对路径
   HDFS权限域linux的权限等完全相同
```

十:HDFS初体验

```
创建文件夹:
`hadoop fs -mkdir /abc`

上传文件(Linux --> HDFS):
`hadoop fs -put  /opt/a.txt  /abc`

查看文件内容:
`hadoop fs -ls /abc`

```

```
下载文件(HDFS --> Linux):
`hadoop fs -get  /abc/a.txt  /opt`

```