

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	https://blog.csdn.net/weixin_44318830	

```
1 三台机器准备工作：
2 1.关闭防火墙      service iptables stop
3                   chkconfig iptables stop
4 2.关闭selinux
5 3.修改主机名
6 4.ssh无密码拷贝数据
7 特别说明(在主节点无密码访问到从节点)
8 ssh-keygen
9 ssh-copy-id 192.168.100.100
10 ssh-copy-id 192.168.100.110
11 ssh-copy-id 192.168.100.120
12 5.设置主机名和IP对应
13 vi /etc/hosts
14 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

```
1 mkdir -p /export/softwares 存放软件压缩包
2 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:

```
1 cd /export/softwares/
2 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.tar.gz
3 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:

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

```
[root@node01 bin]# ./hadoop checknative
19/06/14 08:16:20 INFO bzip2.Bzip2Factory: Successfully loaded & initialized native-bzip2 library system-native
19/06/14 08:16:20 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
Native library checking:
hadoop: true /export/servers/hadoop-2.6.0-cdh5.14.0/lib/native/libhadoop.so.1.0.0
zlib: true /lib64/libz.so.1
snappy: true /usr/lib64/libsnappy.so.1
lz4: true revision:10301
bzip2: true /lib64/libbz2.so.1
openssl: false !cannot load libcrypto.so (libcrypto.so: cannot open shared object file: No such file or directory)!
```

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。

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

```
[root@node01 bin]# ./hadoop checknative
19/06/14 08:16:47 INFO bzip2.Bzip2Factory: Successfully loaded & initialized native-bzip2 library system-native
19/06/14 08:16:47 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
Native library checking:
hadoop: true /export/servers/hadoop-2.6.0-cdh5.14.0/lib/native/libhadoop.so.1.0.0
zlib: true /lib64/libz.so.1
snappy: true /usr/lib64/libsnappy.so.1
lz4: true revision:10301
bzip2: true /lib64/libbz2.so.1
openssl: true /usr/lib64/libcrypto.so
```

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

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

将以下内容添至xml文件指定位置处

```
1 <configuration>
2   <property>
3     <name>fs.defaultFS</name>
4     <value>hdfs://node01:8020</value>
5   </property>
6   <property>
7     <name>hadoop.tmp.dir</name>
8     <value>/export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/tempDatas</value>
9   </property>
10  <!-- 缓冲区大小，实际工作中根据服务器性能动态调整 -->
11  <property>
12    <name>io.file.buffer.size</name>
13    <value>4096</value>
14  </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:

```

<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</value>
  </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>

  <!-- 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 -->
  <property>
    <name>fs.trash.interval</name>
    <value>10080</value>
  </property>
</configuration>

```

该台服务器的主机名

这部分的路径填hadoop的安装路径

https://blog.csdn.net/weixin_44318830

2.修改hdfs-site.xml

第一台机器执行以下命令

```

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

```

将以下内容添至xml文件中

```
1 <configuration>
2   <!-- 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
3   <!--    集群动态上下线
4   <property>
5       <name>dfs.hosts</name>
6       <value>/export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/accept_hc
7   </property>
8
9   <property>
10      <name>dfs.hosts.exclude</name>
11      <value>/export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/deny_host
12  </property>
13      -->
14
15      <property>
16          <name>dfs.namenode.secondary.http-address</name>
17          <value>node01:50090</value>
18      </property>
19
20      <property>
21          <name>dfs.namenode.http-address</name>
22          <value>node01:50070</value>
23      </property>
24      <property>
25          <name>dfs.namenode.name.dir</name>
26          <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/r
27      </property>
28      <!-- 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
29      <property>
30          <name>dfs.datanode.data.dir</name>
31          <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/c
32      </property>
33
34      <property>
35          <name>dfs.namenode.edits.dir</name>
36          <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/c
37      </property>
38      <property>
39          <name>dfs.namenode.checkpoint.dir</name>
40          <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/c
```

```

41     </property>
42     <property>
43         <name>dfs.namenode.checkpoint.edits.dir</name>
44         <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/c
45     </property>
46     <property>
47         <name>dfs.replication</name>
48         <value>2</value>
49     </property>
50     <property>
51         <name>dfs.permissions</name>
52         <value>false</value>
53     </property>
54 <property>
55     <name>dfs.blocksize</name>
56     <value>134217728</value>
57 </property>
58 </configuration>

```

注意事项:

```

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, 然后多个目录用, 进行分割 -->
<!-- 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_host</value>
</property>
<property>
    <name>dfs.hosts.exclude</name>
    <value>/export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/deny_host</value>
</property>
-->
<property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>node02:50090</value>
</property>
<property>
    <name>dfs.namenode.http-address</name>
    <value>node02:50070</value>
</property>
<property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/namenodeDatas</value>
</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, 然后多个目录用, 进行分割 -->
<property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/datanodeDatas</value>
</property>
<property>
    <name>dfs.namenode.edits.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/edits</value>
</property>
<property>
    <name>dfs.namenode.checkpoint.dir</name>
    <value>file:///export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/snn/name</value>
</property>

```

主机名

Hadoop安装路径

https://blog.csdn.net/weixin_44318830

<https://blog.csdn.net/hetnzyyyyyy>

3.修改Hadoop-env.sh

第一台机器执行以下命令

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

```
2 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

```
1 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

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

将以下内容添至xml文件指定位置处中

```
1 <configuration>
2   <property>
3     <name>mapreduce.framework.name</name>
4     <value>yarn</value>
5   </property>
6
7   <property>
8     <name>mapreduce.job.ubertask.enable</name>
9     <value>true</value>
10  </property>
11
12  <property>
13    <name>mapreduce.jobhistory.address</name>
14    <value>node01:10020</value>
15  </property>
16
17  <property>
18    <name>mapreduce.jobhistory.webapp.address</name>
19    <value>node01:19888</value>
20  </property>
21 </configuration>
```

注意事项:


```
<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>node02:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>node02:10088</value>
  </property>
</configuration>
```

主机名

https://blog.csdn.net/weixin_44318830

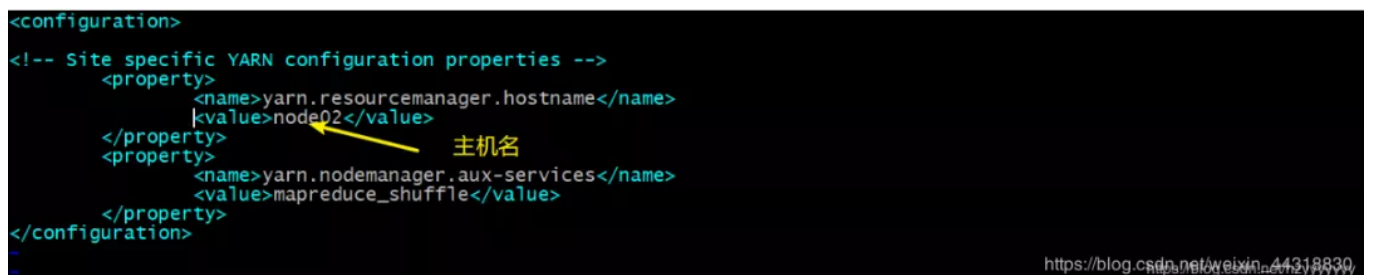
5.修改yarn-site.xml

第一台机器执行以下命令

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

将以下内容添至xml文件指定位置处中

```
1 <configuration>
2   <property>
3     <name>yarn.resourcemanager.hostname</name>
4     <value>node01</value>
5   </property>
6   <property>
7     <name>yarn.nodemanager.aux-services</name>
8     <value>mapreduce_shuffle</value>
9   </property>
10 </configuration>
11
```

注意事项:


```
<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>node02</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

主机名

https://blog.csdn.net/weixin_44318830

6.修改slaves文件

第一台机器执行以下命令

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

进入文件后写上你集群对应的主机名:

```
1 node01
2 node02
3 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

```
1 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/tempDatas
2 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/namenodeData
3 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/datanodeData
4 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/edits
5 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/snn/name
6 mkdir -p /export/servers/hadoop-2.6.0-cdh5.14.0/hadoopDatas/dfs/nn/snn/edits
```

五:安装包的分发

第一台机器执行以下命令

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

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

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

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

配置完成之后生效

```
1 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



```
19/06/14 08:41:20 INFO blockmanagement.BlockManager: maxReplication = 512
19/06/14 08:41:20 INFO blockmanagement.BlockManager: minReplication = 1
19/06/14 08:41:20 INFO blockmanagement.BlockManager: maxReplicationStreams = 1
19/06/14 08:41:20 INFO blockmanagement.BlockManager: replicationCheckInterval = 3000
19/06/14 08:41:20 INFO blockmanagement.BlockManager: encryptDataTransfer = false
19/06/14 08:41:20 INFO blockmanagement.BlockManager: maxNumBlocksToLog = 1000
19/06/14 08:41:20 INFO namenode.FSNamesystem: fsOwner = root (auth:SIMPLE)
19/06/14 08:41:20 INFO namenode.FSNamesystem: supergroup = supergroup
19/06/14 08:41:20 INFO namenode.FSNamesystem: isPermissionEnabled = false
19/06/14 08:41:20 INFO namenode.FSNamesystem: HA Enabled: false
19/06/14 08:41:20 INFO namenode.FSNamesystem: Append Enabled: true
19/06/14 08:41:20 INFO util.GSet: Computing capacity for map InodeMap
19/06/14 08:41:20 INFO util.GSet: VM type = 64-bit
19/06/14 08:41:20 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
19/06/14 08:41:20 INFO util.GSet: capacity = 2A20 = 1048576 entries
19/06/14 08:41:20 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? false
19/06/14 08:41:20 INFO namenode.NameNode: Caching file names occurring more than 10 times
19/06/14 08:41:20 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAccessTimeOnlyChange: false, snapshotDiffAllowSnapRootDesc
19/06/14 08:41:20 INFO util.GSet: Computing capacity for map cachedBlocks
19/06/14 08:41:20 INFO util.GSet: VM type = 64-bit
19/06/14 08:41:20 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
19/06/14 08:41:20 INFO util.GSet: capacity = 2A18 = 262144 entries
19/06/14 08:41:20 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
19/06/14 08:41:20 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
19/06/14 08:41:20 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
19/06/14 08:41:20 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
19/06/14 08:41:20 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
19/06/14 08:41:20 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
19/06/14 08:41:20 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
19/06/14 08:41:20 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
19/06/14 08:41:20 INFO util.GSet: Computing capacity for map NameNodeRetryCache
19/06/14 08:41:20 INFO util.GSet: VM type = 64-bit
19/06/14 08:41:20 INFO util.GSet: 0.02999999932944746% max memory 889 MB = 273.1 KB
19/06/14 08:41:20 INFO util.GSet: capacity = 2A13 = 32768 entries
19/06/14 08:41:20 INFO namenode.FSNamesystem: ACLs enabled? false
19/06/14 08:41:20 INFO namenode.FSNamesystem: XAttrs enabled? true
19/06/14 08:41:20 INFO namenode.FSNamesystem: Maximum size of an xattr: 16384
19/06/14 08:41:20 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1329470727-192.168.100.129-1560526880713
19/06/14 08:41:20 INFO common.Storage: Storage directory /export/servers/Hadoop-2.6.0-cdh5.14.0/HadoopData/namenodeData has been successfully formatted.
19/06/14 08:41:20 INFO common.Storage: Storage directory /export/servers/Hadoop-2.6.0-cdh5.14.0/HadoopData/dfs/nn/edit has been successfully formatted.
19/06/14 08:41:20 INFO namenode.FSImageFormatProtobuf: Saving image file /export/servers/Hadoop-2.6.0-cdh5.14.0/HadoopData/namenodeData/current/fsimage.ckpt
19/06/14 08:41:20 INFO namenode.FSImageFormatProtobuf: Image file /export/servers/Hadoop-2.6.0-cdh5.14.0/HadoopData/namenodeData/current/fsimage.ckpt_00000
19/06/14 08:41:20 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
19/06/14 08:41:20 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at node01/192.168.100.129
*****
```

https://blog.csdn.net/weixin_44318830

启动方式分为三种:

1.单个节点逐一启动

- 1 在主节点上使用以下命令启动 HDFS NameNode: `Hadoop-daemon.sh start namenode`
- 2 在每个从节点上使用以下命令启动 HDFS DataNode: `Hadoop-daemon.sh start datanode`
- 3 在主节点上使用以下命令启动 YARN ResourceManager: `yarn-daemon.sh start resourc`
- 4 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

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

停止集群:

没事不要去停止集群

```
1 sbin/stop-dfs.sh  
2 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

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

一键关闭集群

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

八:浏览器查看启动页面

hdfs集群访问地址：

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

192.168.100.129:50070/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities -

Overview

'node01:8020' (active)

Started:	Fri Jun 14 23:42:39 +0800 2019
Version:	2.6.0-cdh5.14.0, rUnknown
Compiled:	Tue May 08 22:32:00 +0800 2018 by root from Unknown
Cluster ID:	CID-b4c68ccb-bf6c-49e2-949f-c0622910d85d
Block Pool ID:	BP-1329470727-192.168.100.129-1560526880713


Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks = 1 total filesystem object(s).
Heap Memory used 152.05 MB of 266 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 54.5 MB of 55.59 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	135.01 GB
DFS Used:	72 KB (0%)
Non DFS Used:	10.1 GB
DFS Remaining:	118.03 GB (87.42%)
Block Pool Used:	72 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	3 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)

yarn集群访问地址：

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



hadoop

All Applications

Logged in as: drwho

Cluster

About

Nodes

Applications

NEW

NEW PAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	V-Cores Used	V-Cores Total	V-Cores Reserved
0	0	0	0	0	0 B	24 GB	0 B	0	24	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	0	0	0

User Metrics for drwho

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	V-Cores Used	V-Cores Pending	V-Cores Reserved
0	0	0	0	0	0	0 B	0 B	0 B	0 B	0	0	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU V-Cores	Allocated Memory MB	Reserved CPU V-Cores	Reserved Memory MB	Progress	Tracking UI
No data available in table															

Showing 0 to 0 of 0 entries

First Previous Next Last

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]#
```

https://blog.csdn.net/weixin_44318830

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HDFS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!
4. 需要注意的是:
5. 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
6. HDFS权限域linux的权限等完全相同

十:HDFS初体验

- 1 创建文件夹:
- 2 ``hadoop fs -mkdir /abc``
- 3
- 4 上传文件(Linux --> HDFS):
- 5 ``hadoop fs -put /opt/a.txt /abc``
- 6
- 7 查看文件内容:
- 8 ``hadoop fs -ls /abc``
- 9

```
10 下载文件(HDFS --> Linux):  
11 `hadoop fs -get /abc/a.txt /opt`  
12
```