

# Hadoop集群环境搭建

微信公众号：小康新鲜事儿

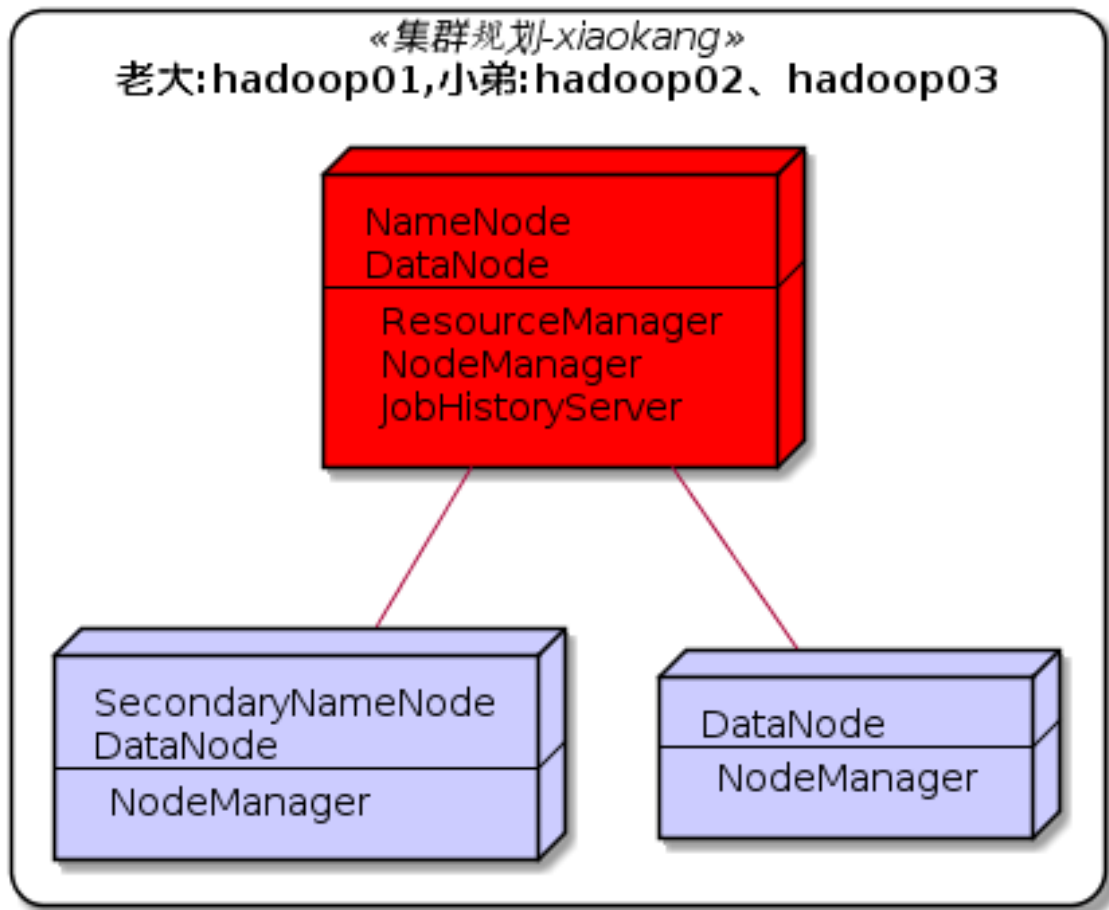


## Hadoop集群环境搭建

- 一、集群规划
- 二、前置条件
- 三、集群配置
  - 3.1 主节点基础网络配置
  - 3.2 配置映射
  - 3.3 根据主节点hadoop01克隆两份系统
  - 3.4 修改各个从节点网络配置
  - 3.5 配置主从节点免密登录
    - 3.5.1 生成密钥
    - 3.5.2 复制公钥
    - 3.5.3 验证免密登录
    - 3.5.4 Gif动图演示
  - 3.6 同步集群时间
    - 3.6.1 配置时间服务器 (hadoop02)
      - 1. 检查ntp包是否安装
      - 2. 设置时间配置文件
      - 3. 设置BIOS与系统时间同步
      - 4. 启动ntp服务并测试
    - 3.6.2 其它节点与时间服务器同步时间
      - 1. 手动同步
      - 2. 定时同步
  - 3.7 修改主节点配置文件
    - 1. hadoop-env.sh
    - 2. core-site.xml
    - 3. hdfs-site.xml
    - 4. mapred-site.xml
    - 5. yarn-site.xml
    - 6. master
    - 7. slaves
  - 3.8 分发程序
  - 3.9 初始化
- 四、启动集群
  - 4.1 查看集群
- 五、提交服务到集群
- 六、基准测试

## 一、集群规划

这里搭建一个 3 节点的 Hadoop 集群，其中三台主机均部署 `DataNode` 和 `NodeManager` 服务，但其中 hadoop02 上部署 `SecondaryNameNode` 服务，hadoop01 上部署 `NameNode`、`ResourceManager` 和 `JobHistoryServer` 服务。



节点ip分配情况如下：

hostname	ip
hadoop01	192.168.239.161
hadoop02	192.168.239.162
hadoop03	192.168.239.163

## 二、前置条件

Hadoop 的运行依赖 JDK，需要预先安装。其安装步骤见：

- [Linux下jdk的安装](#)

## 三、集群配置

### 3.1 主节点基础网络配置

- 固定IP地址
- 配置主机名
- 关闭防火墙

参考视频教程：[Hadoop前置准备](#)

### 3.2 配置映射

配置 ip 地址和主机名映射：

```
[xiaokang@hadoop01 ~]$ sudo vim /etc/hosts
# 文件末尾增加
192.168.239.161 hadoop01
192.168.239.162 hadoop02
192.168.239.163 hadoop03
```

### 3.3 根据主节点hadoop01克隆两份系统

将hadoop01先关机，然后鼠标右击管理->克隆->创建完整克隆->填写名称、选择存放位置即可

### 3.4 修改各个从节点网络配置

修改IP地址和主机名

```
[xiaokang@hadoop01 ~]$ sudo vim /etc/sysconfig/network-scripts/ifcfg-ens33
#将原来的192.168.239.161改为192.168.239.162

#重启网络服务生效
[xiaokang@hadoop01 ~]$ sudo systemctl restart network
```

IP修改完成后，使用XShell进行连接，之后再进行主机名的修改

```
[xiaokang@hadoop01 ~]$ sudo hostname hadoop02

[xiaokang@hadoop01 ~]$ sudo vim /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=hadoop02

[xiaokang@hadoop01 ~]$ sudo vim /etc/hostname
#将原来的hadoop01改为hadoop02
```

退出当前XShell连接，再次登录查看是否配置成功

### 3.5 配置主从节点免密登录

#### 3.5.1 生成密钥

在每台主机上使用 `ssh-keygen` 命令生成公钥私钥对：

```
[xiaokang@hadoop01 ~]$ ssh-keygen -t rsa -C "xiaokang.188@qq.com"
```

#### 3.5.2 复制公钥

将 `hadoop01` 的公钥写到本机和远程机器的 `~/.ssh/authorized_keys` 文件中(另外两台机器上需要做同样的动作)：

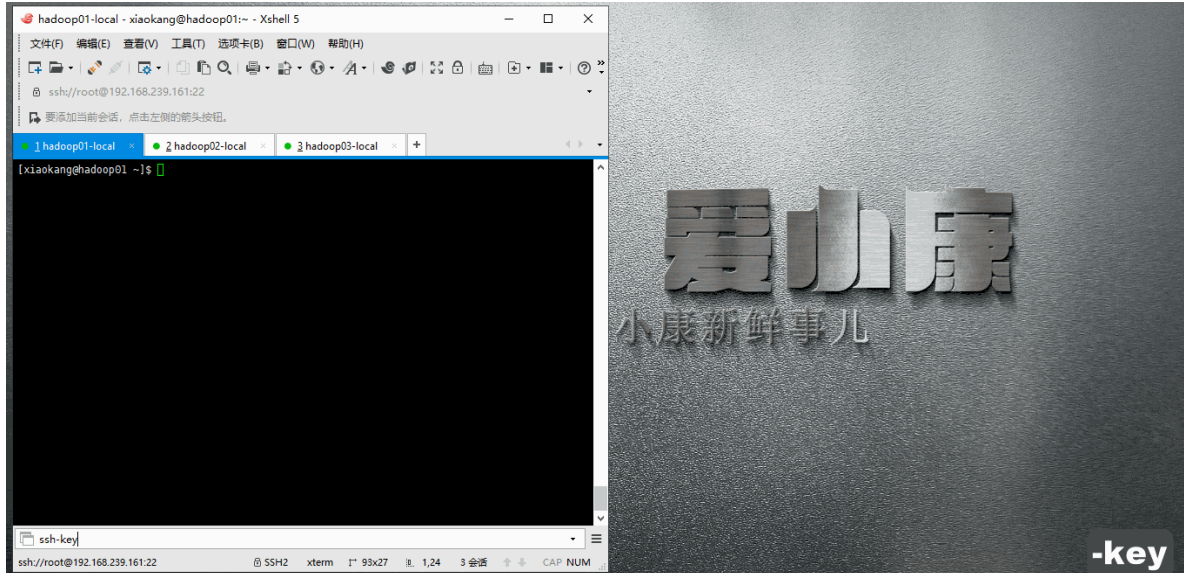
```
[xiaokang@hadoop01 .ssh]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop01
[xiaokang@hadoop01 .ssh]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop02
[xiaokang@hadoop01 .ssh]$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop03
```

#### 3.5.3 验证免密登录

```
ssh hadoop01
ssh hadoop02
ssh hadoop03
```

### 3.5.4 Gif动图演示

[gif动图地址](#)



## 3.6 同步集群时间

关于时间可以采用 `sudo date -s "2020-11-24 11:24:00"` 或 `sudo ntpdate -u ntp.api.bz` 进行设置，但我们一般会采用单独设置一台时间服务器，让其它所有节点与时间服务器的时间进行同步。

### 3.6.1 配置时间服务器 (hadoop02)

#### 1. 检查ntp包是否安装

```
[xiaokang@hadoop02 ~]$ rpm -qa | grep ntp
ntpd-4.2.6p5-29.el7.centos.x86_64
ntp-4.2.6p5-29.el7.centos.x86_64
```

没有安装的话，执行以下命令进行安装：

```
[xiaokang@hadoop02 root]$ sudo yum -y install ntp
```

#### 2. 设置时间配置文件

```
[xiaokang@hadoop02 ~]$ sudo vim /etc/ntp.conf
```

```
#修改一（设置本地网络上的主机不受限制）
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap 为
restrict 192.168.239.0 mask 255.255.255.0 nomodify notrap
#修改二（添加默认的一个内部时钟数据，使用它为局域网用户提供服务）
server 127.127.1.0
fudge 127.127.1.0 stratum 10
#修改三（设置为不采用公共的服务器）
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
```

```
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst 为
#server 0.centos.pool.ntp.org iburst
#server 1.centos.pool.ntp.org iburst
#server 2.centos.pool.ntp.org iburst
#server 3.centos.pool.ntp.org iburst
```

### 3. 设置BIOS与系统时间同步

```
[xiaokang@hadoop02 ~]$ sudo vim /etc/sysconfig/ntp
```

```
#增加如下内容（让硬件时间与系统时间一起同步）
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
SYNC_HWCLOCK=yes
```

### 4. 启动ntp服务并测试

```
[xiaokang@hadoop02 ~]$ sudo systemctl start ntpd
[xiaokang@hadoop02 ~]$ systemctl status ntpd
#设置ntp服务开机自启
[xiaokang@hadoop02 ~]$ sudo systemctl enable ntpd.service

#测试
[xiaokang@hadoop02 ~]$ ntpstat
synchronised to local net (127.127.1.0) at stratum 11
    time correct to within 3948 ms
    polling server every 64 s
[xiaokang@hadoop02 ~]$ sudo ntpq -p
      remote           refid      st t when poll reach   delay    offset   jitter
=====
*LOCAL(0)          .LOCL.          10 l   26   64    3    0.000    0.000    0.000
```

## 3.6.2 其它节点与时间服务器同步时间

先关闭非时间服务器节点的ntpd服务 `sudo systemctl stop ntpd`

#### 1. 手动同步

```
[xiaokang@hadoop03 ~]$ sudo ntpdate hadoop02
24 Nov 11:25:13 ntpdate[2878]: step time server 192.168.239.125 offset
24520304.363894 sec
[xiaokang@hadoop03 ~]$ date
2020年 11月 24日 星期二 11:25:20 CST
```

#### 2. 定时同步

在其他机器配置10分钟与时间服务器同步一次

```
[xiaokang@hadoop03 root]$ sudo vim /etc/crontab
```

编写定时任务如下：

```
*/1 * * * * /usr/sbin/ntpdate hadoop02
```

在03节点的家目录下的xiaokang.txt里面每隔一分钟就追加一个xiaokang的字符串

#加载任务,使之生效

```
[xiaokang@hadoop03 ~]$ sudo crontab /etc/crontab
```

修改时间服务器时间

```
[xiaokang@hadoop02 ~]$ sudo date -s "2020-11-24 11:24:11"
```

十分钟后查看机器是否与时间服务器同步

```
[xiaokang@hadoop03 ~]$ date
```

ps: 测试的时候可以将10分钟调整为1分钟, 节省时间

## 3.7 修改主节点配置文件

先创建好所需目录

```
[xiaokang@hadoop01 hadoop-2.7.7]$ mkdir /opt/software/hadoop-2.7.7/tmp  
[xiaokang@hadoop01 hadoop-2.7.7]$ mkdir -p /opt/software/hadoop-  
2.7.7/dfs/namenode_data  
[xiaokang@hadoop01 hadoop-2.7.7]$ mkdir -p /opt/software/hadoop-  
2.7.7/dfs/datanode_data  
[xiaokang@hadoop01 hadoop-2.7.7]$ mkdir -p /opt/software/hadoop-  
2.7.7/checkpoint/dfs/cname
```

### 1. hadoop-env.sh

```
#25行 export JAVA_HOME  
export JAVA_HOME=/opt/moudle/jdk1.8.0_191  
#33行 export HADOOP_CONF_DIR  
export HADOOP_CONF_DIR=/opt/software/hadoop-2.7.7/etc/hadoop
```

### 2. core-site.xml

```
<configuration>  
  <property>  
    <!--用来指定hdfs的老大, namenode的地址-->  
    <name>fs.defaultFS</name>  
    <value>hdfs://hadoop01:9000</value>  
  </property>  
  <property>  
    <!--用来指定hadoop运行时产生文件的存放目录-->  
    <name>hadoop.tmp.dir</name>  
    <value>file:///opt/software/hadoop-2.7.7/tmp</value>  
  </property>  
  <property>  
    <!--设置缓存大小, 默认4kb-->  
    <name>io.file.buffer.size</name>  
    <value>4096</value>  
  </property>  
</configuration>
```

### 3. hdfs-site.xml

```

<configuration>
  <property>
    <!--数据块默认大小128M-->
    <name>dfs.block.size</name>
    <value>134217728</value>
  </property>
  <property>
    <!--副本数量，不配置的话默认为3-->
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <!--定点检查-->
    <name>fs.checkpoint.dir</name>
    <value>file:///opt/software/hadoop-
2.7.7/checkpoint/dfs/cname</value>
  </property>
  <property>
    <!--namenode节点数据（元数据）的存放位置-->
    <name>dfs.name.dir</name>
    <value>file:///opt/software/hadoop-2.7.7/dfs/namenode_data</value>
  </property>
  <property>
    <!--datanode节点数据（元数据）的存放位置-->
    <name>dfs.data.dir</name>
    <value>file:///opt/software/hadoop-2.7.7/dfs/datanode_data</value>
  </property>
  <property>
    <!--指定secondarynamenode的web地址-->
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop02:50090</value>
  </property>
  <property>
    <!--hdfs文件操作权限,false为不验证-->
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
</configuration>

```

## 4. mapred-site.xml

```

<configuration>
  <property>
    <!--指定mapreduce运行在yarn上-->
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <!--配置任务历史服务器IPC-->
    <name>mapreduce.jobhistory.address</name>
    <value>hadoop01:10020</value>
  </property>
  <property>
    <!--配置任务历史服务器web-UI地址-->
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>hadoop01:19888</value>
  </property>

```

```
</property>
</configuration>
```

## 5. yarn-site.xml

```
<configuration>
  <property>
    <!--指定yarn的老大resourcemanager的地址-->
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop01</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>hadoop01:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>hadoop01:8088</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>hadoop01:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>hadoop01:8031</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>hadoop01:8033</value>
  </property>
  <property>
    <!--NodeManager获取数据的方式-->
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <!--开启日志聚集功能-->
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>
  <property>
    <!--配置日志保留7天-->
    <name>yarn.log-aggregation.retain-seconds</name>
    <value>604800</value>
  </property>
</configuration>
```

## 6. master

在当前配置文件目录内是不存在master文件的，我们使用**vim**写入内容到master内保存即可

```
[xiaokang@hadoop01 ~]$ vim master
hadoop01
```

## 7. slaves



配置所有从属节点的主机名或 IP 地址，每行一个。所有从属节点上的 `DataNode` 服务和 `NodeManager` 服务都会被启动。

```
hadoop01
hadoop02
hadoop03
```

## 3.8 分发程序

将 Hadoop 安装包分发到其他两台服务器，分发后建议在这两台服务器上也配置一下 Hadoop 的环境变量。

```
# 将安装包分发到hadoop002
[xiaokang@hadoop01 hadoop]$ sudo scp -r /opt/software/hadoop-2.7.7/
xiaokang@hadoop02:/opt/software/
# 将安装包分发到hadoop003
[xiaokang@hadoop01 hadoop]$ sudo scp -r /opt/software/hadoop-2.7.7/
xiaokang@hadoop03:/opt/software/
```

shell脚本实现

```
scp-config.sh
#!/bin/bash
#description: 节点间复制文件
#author: xiaokang

#首先判断参数是否存在
args=$#
if [ args -eq 0 ];then
    echo "no args"
    exit 1
fi
#获取文件名称
p1=$1
fname=$(basename $p1)
echo fname=$fname
#获取上级目录到绝对路径
pdir=$(cd $(dirname $p1);pwd -P)
echo pdir=$pdir
#获取当前用户名称
user=$(whoami)
#循环分发
for(( host=2;host<4;host++ ));do
    echo "-----hadoop0$host-----"
    scp -r $pdir/$fname $user@hadoop0$host:$pdir
done
echo "-----分发完成-----"
```

## 3.9 初始化

```
[xiaokang@hadoop01 ~]$ hdfs namenode -format
```

# 四、启动集群

在 `hadoop01` 上启动 Hadoop 集群。此时 `hadoop02` 和 `hadoop03` 上的相关服务也会被启动：

```
# 启动dfs服务
[xiaokang@hadoop01 ~]$ start-dfs.sh
# 启动yarn服务
[xiaokang@hadoop01 ~]$ start-yarn.sh
# 启动任务历史服务器
[xiaokang@hadoop01 ~]$ mr-jobhistory-daemon.sh start historyserver
```

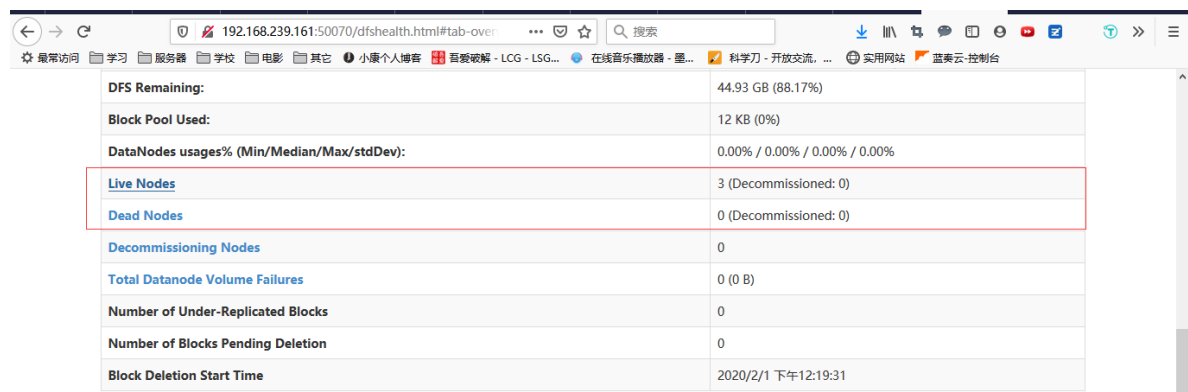
## 4.1 查看集群

在每台服务器上使用 `jps` 命令查看服务进程，或直接进入 Web-UI 界面进行查看，端口为 `50070`。可以看到此时有三个可用的 `Datanode`：

```
[xiaokang@hadoop01 hadoop-2.7.7]$ jps
13664 ResourceManager
13250 NameNode
14101 JobHistoryServer
14135 Jps
13387 DataNode
13773 NodeManager

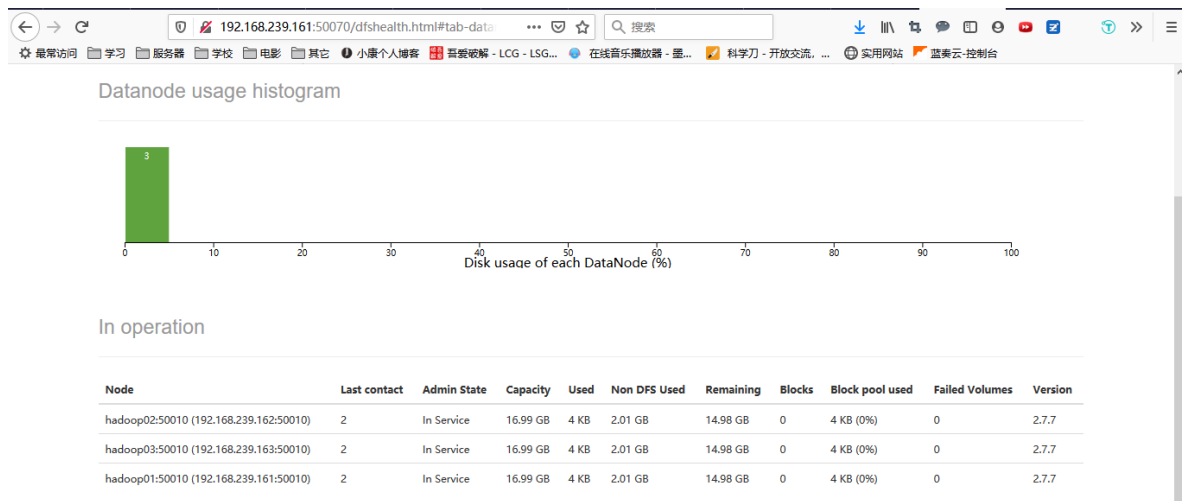
[xiaokang@hadoop02 hadoop-2.7.7]$ jps
9793 SecondaryNameNode
9685 DataNode
9898 NodeManager
9930 Jps

[xiaokang@hadoop03 hadoop-2.7.7]$ jps
9880 DataNode
10024 NodeManager
10056 Jps
```



DFS Remaining:	44.93 GB (88.17%)
Block Pool Used:	12 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
<a href="#">Live Nodes</a>	3 (Decommissioned: 0)
<a href="#">Dead Nodes</a>	0 (Decommissioned: 0)
<a href="#">Decommissioning Nodes</a>	0
<a href="#">Total Datanode Volume Failures</a>	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2020/2/1 下午12:19:31

点击 `Live Nodes` 进入，可以看到每个 `DataNode` 的详细情况：



接着可以查看 Yarn 的情况，端口号为 8088：

Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	24 GB	0 B	0	24	0	3	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Showing 1 to 3 of 3 entries

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack		RUNNING	hadoop02:45747	hadoop02:8042	星期六 二月 01 12:28:26 +0800 2020		0	0 B	8 GB	0	8	2.7.7
/default-rack		RUNNING	hadoop03:42678	hadoop03:8042	星期六 二月 01 12:28:26 +0800 2020		0	0 B	8 GB	0	8	2.7.7
/default-rack		RUNNING	hadoop01:35667	hadoop01:8042	星期六 二月 01 12:28:31 +0800 2020		0	0 B	8 GB	0	8	2.7.7

## 五、提交服务到集群

提交作业到集群的方式和单机环境完全一致，这里以提交 Hadoop 内置的计算 Pi 的示例程序为例，在任何一个节点上执行都可以，命令如下：

```
#第1个11指的是要运行11次map任务
#第2个数字指的是每个map任务，要投掷多少次
[xiaokang@hadoop03 ~]$ hadoop jar /opt/software/hadoop-2.7.7/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar pi 11 24
```

Web-UI界面刚才执行的任务状况：

192.168.239.161:8088/cluster/apps/RUNNING

# hadoop RUNNING Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

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	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
2	0	1	1	12	13 GB	24 GB	0 B	12	24	0	3	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking URL
application_1580532671107_0002	root	QuasiMonteCarlo	MAPREDUCE	default	Sat Feb 1 12:55:27 +0800 2020	N/A	RUNNING	UNDEFINED		ApplicationView

Showing 1 to 1 of 1 entries

First Previous

最终计算结果:

hadoop03-local - xiaokang@hadoop03:/opt/moudle/hadoop-2.7.7/sbin - Xshell 5

ssh://root@192.168.239.163:22

要添加当前会话，点击左侧的箭头按钮。

1 hadoop01-local x 2 hadoop02-local x 3 hadoop03-local x +

```
Reduce input groups=2
Reduce shuffle bytes=308
Reduce input records=22
Reduce output records=0
Spilled Records=44
Shuffled Maps =11
Failed Shuffles=0
Merged Map outputs=11
GC time elapsed (ms)=26994
CPU time spent (ms)=21320
Physical memory (bytes) snapshot=1367105536
Virtual memory (bytes) snapshot=24924368896
Total committed heap usage (bytes)=1362468864

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=1298
File Output Format Counters
Bytes Written=97
Job Finished in 116.884 seconds
Estimated value of Pi is 3.1363636363636364
[xiaokang@hadoop03 sbin]$
```

将文本发送到全部Xshell窗口

ssh://root@192.168.239.163:22 SSH2 xterm 93x27 27,27 3 会话 CAP NUM

## 六、基准测试

#写

```
[xiaokang@hadoop01 ~]$ hadoop jar /opt/software/hadoop-2.7.7/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.7-tests.jar TestDFSIO -write -nrFiles 10 -size 100MB
```

#读

```
[xiaokang@hadoop01 ~]$ hadoop jar /opt/software/hadoop-2.7.7/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.7-tests.jar TestDFSIO -read -nrFiles 10 -size 100MB
```

#清理

```
[xiaokang@hadoop01 ~]$ hadoop jar /opt/software/hadoop-2.7.7/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.7.7-tests.jar TestDFSIO -clean
```