# HDFS 分布式部署

版本：apache-hadoop-2.9.2.tar.gz

## 节点规划

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **主机名** | **IP地址** | **HDFS部署** | **YARN部署** | **ZooKeeper部署** |  |
| hadoop-1 | 192.168.198.224 | namenode  datanode | resource manager  node manager |  |  |
| hadoop-2 | 192.168.198.225 | secondary namenode  datanode | node manager |  |  |
| hadoop-3 | 192.168.198.226 | datanode | node manager |  |  |

## 基础环境配置

注意：以下操作在每个节点都需要执行。

1. 配置hostname：hadoop-1，hadoop-2，hadoop-3
2. 配置/etc/hosts

# /etc/hosts

192.168.198.224 hadoop-1

192.168.198.225 hadoop-2

192.168.198.226 hadoop-3

1. 安装java基础环境：

# yum install java-1.8.0

1. 下载Hadoop安装包，并进行解压到/usr/local目录

# cd /usr/local

# tar -xvf /root/hadoop-2.9.2.tar.gz

1. 配置基础环境变量

# vim /etc/profile

# export PATH=/usr/local/hadoop-xxx/bin:$PATH

# export JAVA\_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64

# hadoop -h

Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]

CLASSNAME run the class named CLASSNAME

# java -version

openjdk version "1.8.0\_191"

OpenJDK Runtime Environment (build 1.8.0\_191-b12)

OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)

1. 基于本地文件系统运行MapReduce程序，检测Java运行环境

# mkdir /root/input

# cp etc/hadoop/\*.xml /root/input

# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar grep /root/input /root/output 'dfs[a-z.]+'

# cat /root/output/\_SUCCESS

# cat /root/output/part-r-00000

# rm -rf /root/input

# rm -rf /root/output

## HDFS配置、运行和查看

**待修改配置文件：**

* **etc/hadoop/core-site.xml**
* **etc/hadoop/hdfs-site.xml**
* **etc/hadoop/slaves**
* **etc/hadoop/hadoop-env.sh**

**# vim etc/hadoop/core-site.xml**

<property>

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

<value>**hdfs://hadoop-1:9000**</value>

</property>

<property>

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

<value>**/opt/hadoop-${user.name}**</value>

</property>

**# vim etc/hadoop/hdfs-site.xml**

<property>

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

<value>**3**</value>

</property>

<property>

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

<value>**hadoop-2:50090**</value>

</property>

**# 增加3个slave节点，运行datanode**

**# vim etc/hadoop/slaves**

hadoop-1

hadoop-2

hadoop-3

**# vim etc/hadoop/hadoop-env.sh**

# The java implementation to use.

export JAVA\_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64

**配置namenode到其他节点的ssh免密码登录：**

# ssh-keygen -t rsa

# ssh-copy-id hadoop-1

# ssh-copy-id hadoop-2

# ssh-copy-id hadoop-3

**同步配置文件到其他node：**

# scp etc/hadoop/\* [root@hadoop-2:/usr/local/hadoop-2.9.2/etc/hadoop/](mailto:root@hadoop-2:/usr/local/hadoop-2.9.2/etc/hadoop/)

# scp etc/hadoop/\* [root@hadoop-3:/usr/local/hadoop-2.9.2/etc/hadoop/](mailto:root@hadoop-3:/usr/local/hadoop-2.9.2/etc/hadoop/)

**格式化HDFS文件系统：**

# ./bin/hdfs namenode -format

# ll /opt/hadoop-root/dfs/name/current/

total 16

-rw-r--r-- 1 root root 323 Dec 29 06:55 fsimage\_0000000000000000000

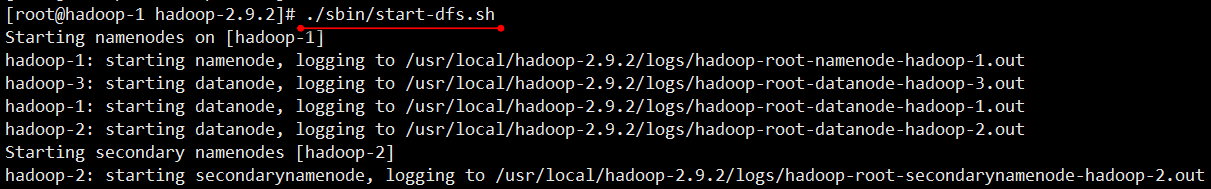
-rw-r--r-- 1 root root 62 Dec 29 06:55 fsimage\_0000000000000000000.md5

-rw-r--r-- 1 root root 2 Dec 29 06:55 seen\_txid

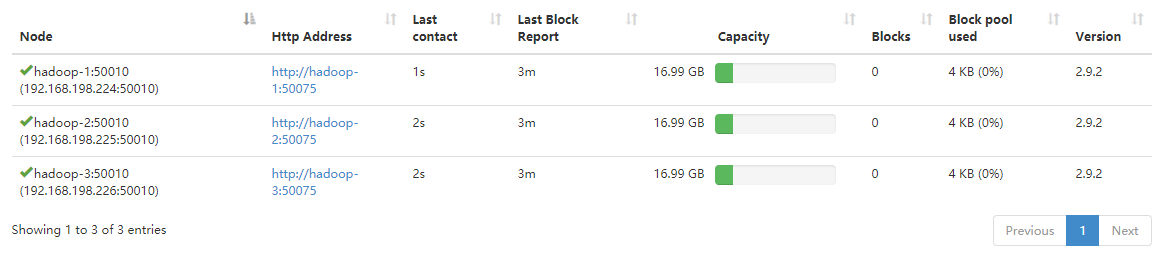
-rw-r--r-- 1 root root 220 Dec 29 06:55 VERSION

**启动HDFS进程**

**# ./sbin/start-dfs.sh**



访问<http://192.168.198.224:50070>，显示DataNode数量和状态正常



## MapReduce程序测试

# 在HDFS中准备用于MapReduce程序处理的输入文件

# hdfs dfs -mkdir -p /user/test/input

# hdfs dfs -put etc/hadoop/\* /user/test/input

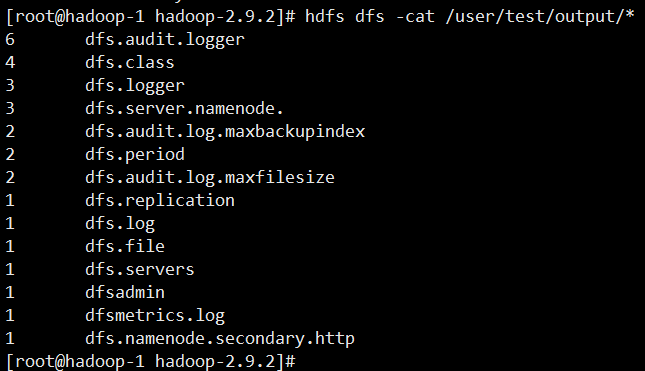
# hdfs dfs -ls /user/test/input

# 运行MapReduce程序

# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar grep /user/test/input /user/test/output 'dfs[a-z.]+'

# 检测运行结果

# hdfs dfs -cat /user/test/output/\*



## 基于YARN框架执行MapReduce程序

**# 配置MapReduce基于YARN调度框架**

**# 配置Job History Server**

**# cp etc/hadoop/mapred-site.xml.template etc/hadoop/mapred-site.xml**

**# vim etc/hadoop/mapred-site.xml**

<property>

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

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

</property>

<property>

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

<value>**hadoop-1:10020**</value>

</property>

<property>

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

<value>**hadoop-1:19888**</value>

</property>

**# vim etc/hadoop/yarn-site.xml**

<property>

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

<value>**hadoop-1**</value>

</property>

<property>

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

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

</property>

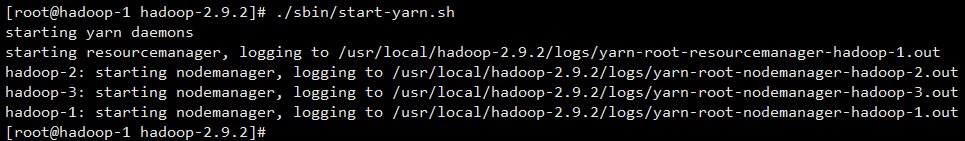
**完成配置更新后，同步到其他node：**

# scp etc/hadoop/\* [root@hadoop-2:/usr/local/hadoop-2.9.2/etc/hadoop/](mailto:root@hadoop-2:/usr/local/hadoop-2.9.2/etc/hadoop/)

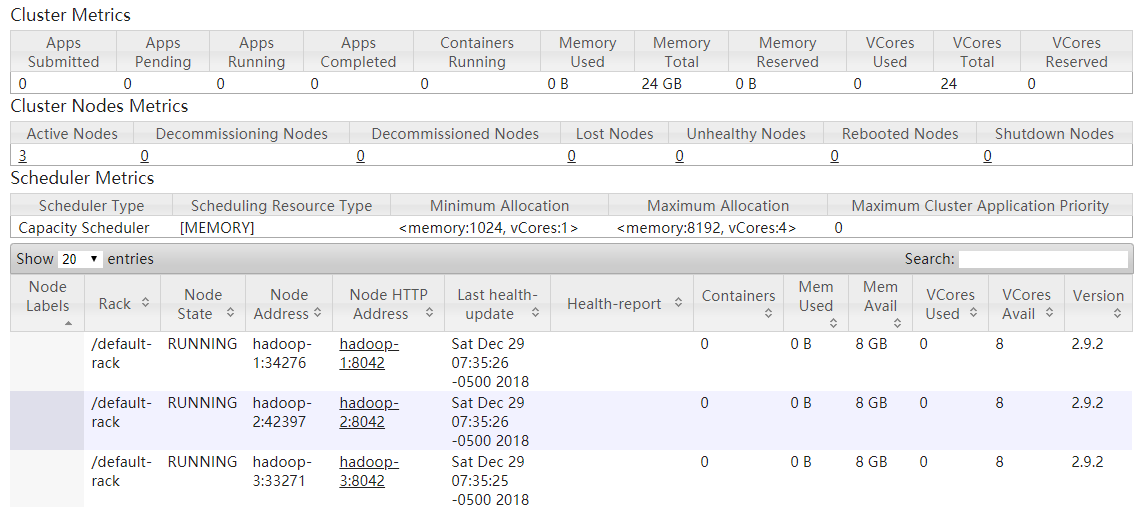
# scp etc/hadoop/\* [root@hadoop-3:/usr/local/hadoop-2.9.2/etc/hadoop/](mailto:root@hadoop-3:/usr/local/hadoop-2.9.2/etc/hadoop/)

**# 启动YARN框架**

# ./sbin/start-yarn.sh



**访问**[**http://192.168.198.224:8088**](http://192.168.198.224:8088)**，显示NodeManager数量和状态正常**

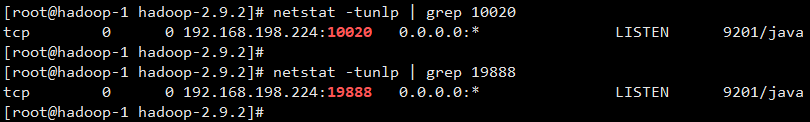


**# 启动HDFS**

# ./sbin/start-dfs.sh

**# 启动Job History Server**

# ./sbin/mr-jobhistory-daemon.sh start historyserver

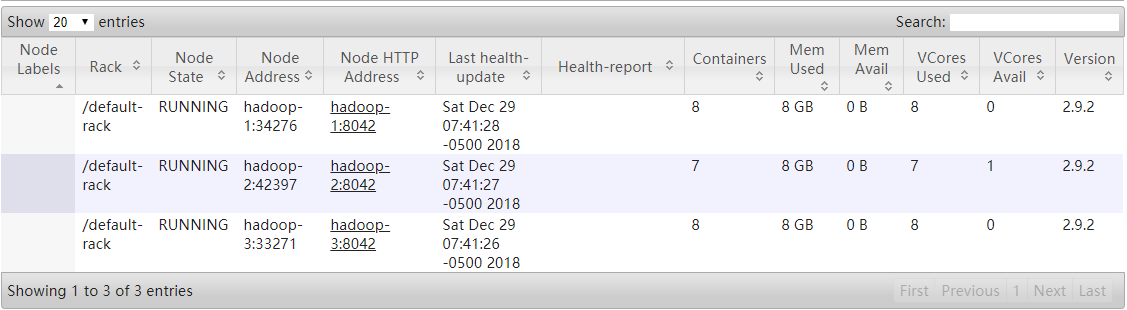


**# 基于YARN框架调度和执行MapReduce程序**

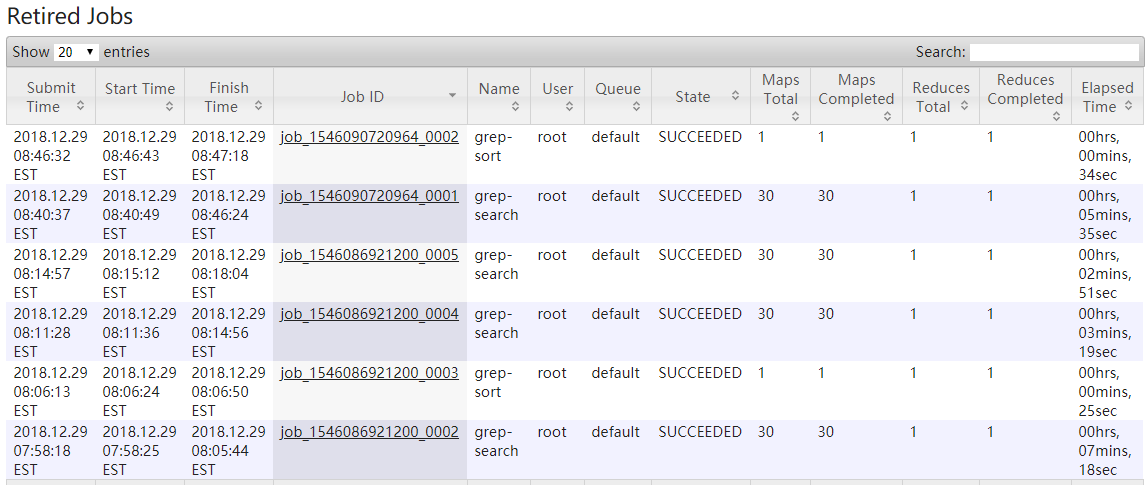
# hdfs dfs -rm -r /user/test/output

# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar grep /user/test/input /user/test/output 'dfs[a-z.]+'

**# 查看MapReduce任务运行状态（4G内存的虚拟机，运行超级慢）**



**访问**[**http://192.168.198.244:19888**](http://192.168.198.244:19888)**，查看job history信息**

****

备注：job history是一个单一的历史服务器，不需要在集群中启动多个服务。使用时，需要在配置文件中指定一个服务器，然后启动job history server服务。

疑问：如果未启动job history server，运行mapreduce出现连接job history server失败的提示：

“Retrying connect to server: 0.0.0.0/0.0.0.0:10020.”

## start和stop初步总结

**start：**

* ./sbin/start-dfs.sh
* ./sbin/start-yarn.sh
* ./sbin/mr-jobhistory-daemon.sh start historyserver

**stop:**

* ./sbin/stop-dfs.sh
* ./sbin/stop-yarn.sh
* ./sbin/mr-jobhistory-daemon.sh stop historyserver

HDFS：<http://192.168.198.224:50070>

YARN：<http://192.168.198.224:8088>

JobHistory：<http://192.168.198.224:19888>

# 基于Python接口的HDFS二次开发

安装hdfs包：

# yum install epel-release

# yum install python-pip

# pip install hdfs

一个简单的例子：通过python代码访问HDFS

from hdfs.client import Client

client = Client("http://hadoop-1:50070")

for item in client.list("/"):

print item

# output

tmp

user

一个简单的例子：通过CLI访问HDFS

[root@hadoop-1 ~]# **cat /root/.hdfscli.cfg**

**[global]**

**default.alias = hadoop**

**[hadoop.alias]**

**url = http://hadoop-1:50070**

**user = root**

[root@hadoop-1 ~]#

[root@hadoop-1 ~]# **hdfscli --alias=hadoop**

Welcome to the interactive HDFS python shell.

The HDFS client is available as `CLIENT`.

>>> **CLIENT.list("/")**

[u'tmp', u'user']

>>>

>>> exit()

[root@hadoop-1 ~]#