基于 Docker 构建 Hadoop 平台

----- written by Mike.Wang 2022.07.15

0. 绪论

使用Docker搭建Hadoop技术平台,包括安装Docker、Java、Scala、Hadoop、 Hbase、Spark。

集群共有5台机器, 主机名分别为 h01、h02、h03、h04、h05。其中 h01 为 master, 其他的为 slave。

虚拟机配置:建议1盒2线程、8G内存、30G硬盘。最早配置4G内存,HBase和Spark运行异常。

- JDK 1.8
- Scala 2.11.12
- Hadoop 3.3.3
- Hbase 3.0.0
- Spark 3.3.0

1. Docker

1.1 Ubuntu 22.04 安装Docker

在 Ubuntu 下对 Docker 的操作都需要加上 sudo ,如果已经是 root 账号了,则不需要。 如果不加 sudo ,Docker 相关命令会无法执行。

在 Ubuntu 下安装 Docker 的时候需在管理员的账号下操作。

mike@ubuntu2204:~\$ wget -q0- https://get.docker.com/ | sh

安装完成之后,以 sudo 启动 Docker 服务。

mike@ubuntu2204:~\$ sudo service docker start

显示 Docker 中所有正在运行的容器,由于 Docker 才安装,我们没有运行任何容器,所以显示结果如下所示。

mike@ubuntu2204:~\$ sudo docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES mike@ubuntu2204:~\$

1.2 使用Docker

现在的 Docker 网络能够提供 DNS 解析功能,我们可以使用如下命令为接下来的 Hadoop 集群单独构 建一个虚拟的网络。可以采用直通、桥接或macvlan方式,这里采用桥接模式,可以做到5台主机互联, 并能访问宿主机和网关,可以连接外网,便于在线下载程序资源。

mike@ubuntu2204:~\$sudo docker network create --driver=bridge hadoop

以上命令创建了一个名为 hadoop 的虚拟桥接网络,该虚拟网络内部提供了自动的DNS解析服务。使用 下面这个命令查看 Docker 中的网络,可以看到刚刚创建的名为 hadoop 的虚拟桥接网络。

mike@ubuntu220	4:~\$ sudo	docker net	work ls	
[sudo] passwor	d for mike	9:		
NETWORK ID	NAME	DRIVER	SCOPE	
3948edc3e8f3	bridge	bridge	local	
337965dd9b1e	hadoop	bridge	local	
cb8f2c453adc	host	host	local	
fff4bd1c15ee	mynet	macvlan	local	
30e1132ad754	none	null	local	
mike@ubuntu220	4:~\$			

查找 ubuntu 容器

打开<u>https://hub.docker.com/</u>官网,搜索ubuntu,找到官方认证镜像,这里选取第一个

← → C ■ hub.docker.com/search?q=ubuntu	DOWNLC	DAD VIDEO X
M Gmail 💶 YouTube 💡 地图 💺 翻译 🛃 在线节拍器 Mus 🛅 操作系统(双语) 🤇	(26条消息) manja 🧲 (27条消息) Manja 🧲 (27条消息) vim 配 🖓 manjaro与pytho 课程中心	辽宁省 🎧 Issues・chxuan/v 💽 (27条
📥 docker hub 🔍 ubuntu	Explore Repositories Organizations Help -	Upgrade 🍈 wangdg68 🗸
Filters Products Images Plugins Trusted Content Q Docker Official Image @ C & Verified Publisher @	1 - 25 of 124,530 results for ubuntu.	Best Match Herein Best Match Best Matc
 Open Source Program Operating Systems Linux Windows 	websphere-liberty DOCKER OFFICIAL IMAGE Updated 20 days ago Websphere Liberty multi-architecture images based on Ubuntu 18.04 Linux IBM Z 386 x86-64 PowerPC 64 LE	10M+ 286 Downloads Stars
Architectures ARM ARM ARM BARM BARM BARM BARM BARM BA	open-liberty Q DOCKER OFFICIAL IMAGE Updated 22 days ago Open Liberty multi-architecture images based on Ubuntu 18.04 Linux PowerPC 64 LE IBM Z 386 x86-64	10M+ 53 Downloads Stars
PowerPC 64 LE x86 x86-64	Ubuntu-debootstrap Q DOCKER OFFICIAL IMAGE Updated syears ago DEPRECATED; use "ubuntu" Instead	5M+ 46 Downloads Stars

点击第一个ubuntu,查找可选用的版本,这里选取22.04

Quick reference

- Maintained by: Canonical and Tianon (Debian Developer)
- Where to get help: the Docker Community Forums, the Docker Community Slack, or Stack Overflow

Supported tags and respective Dockerfile links

- 18.04 , bionic-20220531 , bionic
- 20.04 , focal-20220531 , focal
- 21.10 , impish-20220531 , impish
- 22.04 , jammy-20220531 , jammy , latest , rolling
- 22.10 , kinetic-20220602 , kinetic , devel
- 14.04 , trusty-20191217 , trusty
- 16.04 , xenial-20210804 , xenial

Quick reference (cont.)

```
下载 ubuntu 22.04 版本的镜像文件
```

```
mike@ubuntu2204:~$ sudo docker pull ubuntu:22.04
```

查看已经下载的镜像

```
mike@ubuntu2204:~$ sudo docker images
[sudo] password for mike:
REPOSITORY TAG IMAGE ID CREATED SIZE
newuhadoop latest fe08b5527281 3 days ago 2.11GB
ubuntu 22.04 27941809078c 6 weeks ago 77.8MB
mike@ubuntu2204:~$
```

根据镜像启动一个容器,可以看出 shell 已经是容器的 shell 了,这里注意@后面的容器ID与上图镜像ID 一致

mike@ubuntu2204:~\$ sudo docker run -it ubuntu:22.04 /bin/bash root@27941809078c:/#

输入 exit 可以退出容器,不过建议使用 Ctrl + P + Q,退出容器状态,但仍让容器处于后台运行状态。

mike@ubuntu2204:~\$

查看本机上所有的容器

mike@ubuntu220 [sudo] passwor	4:~\$ sudo doc d for mike:	ker ps -a			
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
					NAMES
8016da5278ae	newuhadoop	"/bin/bash"	3 days ago	Up 2 days	
					h05
409c7e8aa2e9	newuhadoop	"/bin/bash"	3 days ago	Up 2 days	
					h04
0d8af236e1e7	newuhadoop	"/bin/bash"	3 days ago	Up 2 days	
					h03
72d62b7d4874	newuhadoop	"/bin/bash"	3 days ago	Up 2 days	
					h02
d4d3ca3bbb61	newuhadoop	"/bin/bash"	3 days ago	Up 2 days	0.0.0.0:8088-
>8088/tcp, :::	8088->8088/tc	p, 0.0.0.0:987	0->9870/tcp,	:::9870->987	0/tcp h01
mike@ubuntu220	4:~\$				

此处会看到刚刚创建好的容器,并在后台运行。这里因为是后期制作的教程,为了节省内存,只保留了5 台hadoop的容器,最原始的容器已经删除。

启动一个状态为退出的容器,最后一个参数为容器 ID

mike@ubuntu2204:~\$ sudo docker start 27941809078c

进入一个容器

mike@ubuntu2204:~\$ sudo docker attach 27941809078c

关闭一个容器

mike@ubuntu2204:~\$ sudo docker stop 27941809078c

2. 安装集群

主要是安装 JDK 1.8 的环境,因为 Spark 要 Scala, Scala 要 JDK 1.8,以及 Hadoop,以此来构建基础 镜像。

2.1 安装 Java 与 Scala

进入之前的 Ubuntu 容器

先更换 apt 的源

2.1.1 修改 apt 源

备份源

```
root@27941809078c:/# cp /etc/apt/sources.list /etc/apt/sources_init.list
root@27941809078c:/#
```

先删除就源文件,这个时候没有 vim 工具..

root@27941809078c:/# rm /etc/apt/sources.list

复制以下命令,回车,即可一键切换到阿里云 ubuntu 22.04镜像: (此时已经是root权限,提示符为 #)

bash -c "cat << EOF > /etc/apt/sources.list && apt update deb http://mirrors.aliyun.com/ubuntu/ jammy main restricted universe multiverse deb-src http://mirrors.aliyun.com/ubuntu/ jammy main restricted universe multiverse deb http://mirrors.aliyun.com/ubuntu/ jammy-security main restricted universe multiverse deb-src http://mirrors.aliyun.com/ubuntu/ jammy-security main restricted universe multiverse deb http://mirrors.aliyun.com/ubuntu/ jammy-updates main restricted universe multiverse deb-src http://mirrors.aliyun.com/ubuntu/ jammy-updates main restricted universe multiverse deb http://mirrors.aliyun.com/ubuntu/ jammy-proposed main restricted universe multiverse deb-src http://mirrors.aliyun.com/ubuntu/ jammy-proposed main restricted universe multiverse deb http://mirrors.aliyun.com/ubuntu/ jammy-backports main restricted universe multiverse deb-src http://mirrors.aliyun.com/ubuntu/ jammy-backports main restricted universe multiverse EOF"

再使用 apt update / apt upgrade 来更新, update更列表, upgrade更新包

root@27941809078c:/# apt update root@27941809078c:/# apt upgrade

2.1.2 安装 Java与 Scala

安装 jdk 1.8, 直接输入命令

root@27941809078c:/# apt install openjdk-8-jdk

测试一下安装结果

```
root@27941809078c:/# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
root@27941809078c:/#
```

接下来安装scala

root@27941809078c:/# apt install scala

测试一下安装结果

```
root@27941809078c:/# scala
Welcome to Scala 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_312).
Type in expressions for evaluation. Or try :help.
```

scala>

输入:quit 退出scala

2.2 安装 Hadoop

- 在当前容器中将配置配好
- 导入出为镜像
- 以此镜像为基础创建五个容器,并赋予 hostname
- 进入 h01 容器, 启动 Hadoop

2.2.1 安装 Vim 与 网络工具包

安装 vim, 用来编辑文件

root@27941809078c:/# apt install vim

安装 net-tools、iputils-ping、iproute2网络工具包,目的是为了使用ping、ifconfig、ip、traceroute 等命令

```
root@27941809078c:/# apt install net-tools
root@27941809078c:/# apt install iputils-ping
root@27941809078c:/# apt install iproute2
```

2.2.2 安装 SSH

安装 SSH,并配置免密登录,由于后面的容器之间是由一个镜像启动的,就像同一个磨具出来的 5 把锁 与钥匙,可以互相开锁。所以在当前容器里配置 SSH 自身免密登录就 OK 了。

安装 SSH 服务器端

root@27941809078c:/# apt install openssh-server

安装 SSH 的客户端

root@27941809078c:/# apt install openssh-client

进入当前用户的用户根目录

```
root@27941809078c:/# cd ~
root@27941809078c:~#
```

生成密钥,不用输入,一直回车就行,生成的密钥在当前用户根目录下的 .ssh 文件夹中。以 .开头的文件与文件夹 ls 是隐藏的,需要 ls -al 才能查看。

```
root@27941809078c:~# ssh-keygen -t rsa -P ""
```

```
root@27941809078c:~# cat .ssh/id_rsa.pub >> .ssh/authorized_keys
root@27941809078c:~#
```

启动 SSH 服务

```
root@27941809078c:~# service ssh start
 * Starting OpenBSD Secure Shell server sshd
```

[OK] root@27941809078c:~#

免密登录自己

```
root@27941809078c:~# ssh 127.0.0.1
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-41-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
To restore this content, you can run the 'unminimize' command.
Last login: Sun Jul 17 08:26:15 2022 from 172.18.0.1
* Starting OpenBSD Secure Shell server sshd
root@27941809078c:~#
```

修改 .bashrc 文件, 启动 shell 的时候, 自动启动 SSH 服务

用 vim 打开 .bashrc 文件

root@27941809078c:~# vim ~/.bashrc

按一下 i 键, 使得 vim 进入插入模式, 此时终端的左下角会显示为 -- INSERT --, 将光标移动到最后 面, 添加一行 (Caps + g 可直接到最后一行)

service ssh start

添加完的结果为,只显示最后几行

```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi
service ssh start
```

按一下 Esc 键, 使得 vim 退出插入模式

再输入英文模式下的冒号:,此时终端的左下方会有一个冒号:显示出来

再输入三个字符 wq!, 这是一个组合命令

- w是保存的意思
- q 是退出的意思
- !是强制的意思

再输入回车,退出 vim。

此时,SSH 免密登录已经完全配置好。

2.2.3 安装 Hadoop

下载 Hadoop 的安装文件

```
root@27941809078c:~# wget https://mirrors.aliyun.com/apache/hadoop/common/hadoop-
3.3.3/hadoop-3.3.3.tar.gz
```

解压到 /usr/local 目录下面并重命名文件夹

```
root@27941809078c:~# tar -zxvf hadoop-3.3.3.tar.gz -C /usr/local/
root@27941809078c:~# cd /usr/local/
root@27941809078c:/usr/local# mv hadoop-3.3.3 hadoop
root@27941809078c:/usr/local#
```

修改 /etc/profile 文件, 添加一下环境变量到文件中

先用 vim 打开 /etc/profile

vim /etc/profile

追加以下内容

JAVA_HOME 为 JDK 安装路径,使用 apt 安装就是这个,用 update-alternatives --config java 可 查看

```
#java
```

export	JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export	JRE_HOME=\${JAVA_HOME}/jre
export	CLASSPATH=.:\${JAVA_HOME}/lib:\${JRE_HOME}/lib
export	PATH=\${JAVA_HOME}/bin:\$PATH
#hadoop)
export	HADOOP_HOME=/usr/local/hadoop
export	PATH=\$PATH:\$HADOOP_HOME/bin:\$HADOOP_HOME/sbin
export	HADOOP_COMMON_HOME=\$HADOOP_HOME
export	HAD00P_HDFS_H0ME=\$HAD00P_H0ME
export	HADOOP_MAPRED_HOME=\$HADOOP_HOME
export	HADOOP_YARN_HOME=\$HADOOP_HOME
export	HAD00P_INSTALL=\$HAD00P_H0ME
export	HADOOP_COMMON_LIB_NATIVE_DIR=\$HADOOP_HOME/lib/native
export	HAD00P_CONF_DIR=\$HAD00P_HOME
export	HADOOP_LIBEXEC_DIR=\$HADOOP_HOME/libexec
export	JAVA_LIBRARY_PATH=\$HADOOP_HOME/lib/native:\$JAVA_LIBRARY_PATH
export	HADOOP_CONF_DIR=\$HADOOP_PREFIX/etc/hadoop

```
export HDFS_DATANODE_USER=root
export HDFS_DATANODE_SECURE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export HDFS_NAMENODE_USER=root
export YARN_RESOURCEMANAGER_USER=root
export YARN_NODEMANAGER_USER=root
```

使环境变量生效

root@27941809078c:/usr/local# source /etc/profile root@27941809078c:/usr/local#

在目录 /usr/local/hadoop/etc/hadoop 下,修改6个重要配置文件

修改 hadoop-env.sh 文件,在文件末尾添加一下信息

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HDFS_NAMENODE_USER=root
export HDFS_DATANODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export YARN_RESOURCEMANAGER_USER=root
export YARN_NODEMANAGER_USER=root

修改 core-site.xml, 修改为

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://h01:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/hadoop3/hadoop/tmp</value>
</property>
</configuration>
```

修改 hdfs-site.xml, 修改为

```
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/home/hadoop3/hadoop/hdfs/name</value>
</property>
<property>
<name>dfs.namenode.data.dir</name>
<value>/home/hadoop3/hadoop/hdfs/data</value>
</property>
</configuration>
```

修改 mapred-site.xml, 修改为

<configuration></configuration>
<property></property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
<property></property>
<name>mapreduce.application.classpath</name>
<value></value>
/usr/local/hadoop/etc/hadoop,
/usr/local/hadoop/share/hadoop/common/*,
/usr/local/hadoop/share/hadoop/common/lib/*,
/usr/local/hadoop/share/hadoop/hdfs/*,
/usr/local/hadoop/share/hadoop/hdfs/lib/*,
/usr/local/hadoop/share/hadoop/mapreduce/*,
/usr/local/hadoop/share/hadoop/mapreduce/lib/*,
/usr/local/hadoop/share/hadoop/yarn/*,
/usr/local/hadoop/share/hadoop/yarn/lib/*

修改 yarn-site.xml,修改为

<configuration></configuration>
<property></property>
<name>yarn.resourcemanager.hostname</name>
<value>h01</value>
<property></property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>

修改 worker 为

h01 h02 h03 h04 h05

此时, hadoop已经配置好了

2.2.4 在 Docker 中启动集群

先将当前容器导出为镜像,并查看当前镜像。使用 ctrl + p + q,退出容器,回到宿主机

```
mike@ubuntu2204:~$ sudo docker commit -m "hadoop" -a "hadoop" 27941809078c
newuhadoop
sha256:648d8e082a231919faeaa14e09f5ce369b20879544576c03ef94074daf978823
mike@ubuntu2204:~$ sudo docker images
[sudo] password for mike:
REPOSITORY
           TAG
                                    CREATED
                                                 SIZE
                    IMAGE ID
newuhadoop
          latest fe08b5527281 4 days ago
                                                 2.11GB
            22.04
ubuntu
                    27941809078c 6 weeks ago 77.8MB
mike@ubuntu2204:~$
```

启动5个终端,分别执行这几个命令

第一条命令启动的是 ho1 是做 master 节点的,所以暴露了端口,以供访问 web 页面

```
mike@ubuntu2204:~$ sudo docker run -it --network hadoop -h "h01" --name "h01" -p
9870:9870 -p 8088:8088 newuhadoop /bin/bash
 * Starting OpenBSD Secure Shell server sshd
       [ OK ]
root@h01:/#
```

其余的四条命令就是几乎一样的了,注意:启动容器后,使用 ctrl + p + q 退回到宿主机,之后再启动下 一个容器

```
mike@ubuntu2204:~$ sudo docker run -it --network hadoop -h "h02" --name "h02"
newuhadoop /bin/bash
[sudo] password for mike:
* Starting OpenBSD Secure Shell server sshd
       [ OK ]
root@h02:/#
mike@ubuntu2204:~$ sudo docker run -it --network hadoop -h "h03" --name "h03"
newuhadoop /bin/bash
[sudo] password for mike:
 * Starting OpenBSD Secure Shell server sshd
       [ OK ]
root@h03:/#
mike@ubuntu2204:~$ sudo docker run -it --network hadoop -h "h04" --name "h04"
newuhadoop /bin/bash
[sudo] password for mike:
* Starting OpenBSD Secure Shell server sshd
       [ OK ]
root@h04:/#
mike@ubuntu2204:~$ sudo docker run -it --network hadoop -h "h05" --name "h05"
newuhadoop /bin/bash
[sudo] password for mike:
* Starting OpenBSD Secure Shell server sshd
       [ OK ]
root@h05:/#
```

接下来,在 ho1 主机中,启动 Haddop 集群

先进行格式化操作,不格式化操作,hdfs会起不来

root@h01:/usr/local/hadoop/bin# ./hadoop namenode -format

进入 hadoop 的 sbin 目录

root@h01:/# cd /usr/local/hadoop/sbin/ root@h01:/usr/local/hadoop/sbin#

启动 hadoop

root@h01:/usr/local/hadoop/sbin# ./start-all.sh Starting namenodes on [h01] h01: Warning: Permanently added 'h01,172.18.0.2' (ECDSA) to the list of known hosts. Starting datanodes h05: Warning: Permanently added 'h05,172.18.0.6' (ECDSA) to the list of known hosts. h02: Warning: Permanently added 'h02,172.18.0.3' (ECDSA) to the list of known hosts. h03: Warning: Permanently added 'h03,172.18.0.4' (ECDSA) to the list of known hosts. h04: Warning: Permanently added 'h04,172.18.0.5' (ECDSA) to the list of known hosts. h03: WARNING: /usr/local/hadoop/logs does not exist. Creating. h05: WARNING: /usr/local/hadoop/logs does not exist. Creating. h02: WARNING: /usr/local/hadoop/logs does not exist. Creating. h04: WARNING: /usr/local/hadoop/logs does not exist. Creating. Starting secondary namenodes [h01] Starting resourcemanager Starting nodemanagers root@h01:/usr/local/hadoop/sbin#

使用jps查看集群启动状态(这个状态不是固定不变的,随着应用不同而不同,但至少应该有3个)

root@h01:~# jps 10017 HRegionServer 10609 Master 9778 HQuorumPeer 8245 SecondaryNameNode 8087 DataNode 9881 HMaster 41081 Jps 10684 Worker 7965 NameNode 8477 ResourceManager 8591 NodeManager root@h01:~#

使用命令 ./hdfs dfsadmin -report 可查看分布式文件系统的状态

root@h01:/usr/local/hadoop/bin# ./hdfs dfsadmin -report Configured Capacity: 90810798080 (84.57 GB)

```
Present Capacity: 24106247929 (22.45 GB)
DFS Remaining: 24097781497 (22.44 GB)
DFS Used: 8466432 (8.07 MB)
DFS Used%: 0.04%
Replicated Blocks:
       Under replicated blocks: 0
        Blocks with corrupt replicas: 0
       Missing blocks: 0
       Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
 -----
Live datanodes (5):
Name: 172.18.0.2:9866 (h01)
Hostname: h01
Decommission Status : Normal
Configured Capacity: 18162159616 (16.91 GB)
DFS Used: 2875392 (2.74 MB)
Non DFS Used: 11887669248 (11.07 GB)
DFS Remaining: 4712182185 (4.39 GB)
DFS Used%: 0.02%
DFS Remaining%: 25.95%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 10
Last contact: Wed Jul 20 04:55:01 GMT 2022
Last Block Report: Tue Jul 19 23:36:54 GMT 2022
Num of Blocks: 293
Name: 172.18.0.3:9866 (h02.hadoop)
Hostname: h02
Decommission Status : Normal
Configured Capacity: 18162159616 (16.91 GB)
DFS Used: 1396736 (1.33 MB)
Non DFS Used: 11889147904 (11.07 GB)
DFS Remaining: 4846399828 (4.51 GB)
DFS Used%: 0.01%
DFS Remaining%: 26.68%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 8
Last contact: Wed Jul 20 04:55:01 GMT 2022
Last Block Report: Tue Jul 19 23:51:39 GMT 2022
```

Name: 172.18.0.4:9866 (h03.hadoop) Hostname: h03 Decommission Status : Normal Configured Capacity: 18162159616 (16.91 GB) DFS Used: 1323008 (1.26 MB) Non DFS Used: 11889221632 (11.07 GB) DFS Remaining: 5114835114 (4.76 GB) DFS Used%: 0.01% DFS Remaining%: 28.16% Configured Cache Capacity: 0 (0 B) Cache Used: 0 (0 B) Cache Remaining: 0 (0 B) Cache Used%: 100.00% Cache Remaining%: 0.00% Xceivers: 4 Last contact: Wed Jul 20 04:55:01 GMT 2022 Last Block Report: Wed Jul 20 02:14:39 GMT 2022 Num of Blocks: 151 Name: 172.18.0.5:9866 (h04.hadoop) Hostname: h04 Decommission Status : Normal Configured Capacity: 18162159616 (16.91 GB) DFS Used: 1527808 (1.46 MB) Non DFS Used: 11889016832 (11.07 GB) DFS Remaining: 4712182185 (4.39 GB) DFS Used%: 0.01% DFS Remaining%: 25.95% Configured Cache Capacity: 0 (0 B) Cache Used: 0 (0 B) Cache Remaining: 0 (0 B) Cache Used%: 100.00% Cache Remaining%: 0.00% Xceivers: 10 Last contact: Wed Jul 20 04:55:01 GMT 2022 Last Block Report: Wed Jul 20 00:42:09 GMT 2022 Num of Blocks: 134 Name: 172.18.0.6:9866 (h05.hadoop) Hostname: h05 Decommission Status : Normal Configured Capacity: 18162159616 (16.91 GB) DFS Used: 1343488 (1.28 MB) Non DFS Used: 11889201152 (11.07 GB) DFS Remaining: 4712182185 (4.39 GB) DFS Used%: 0.01% DFS Remaining%: 25.95% Configured Cache Capacity: 0 (0 B) Cache Used: 0 (0 B) Cache Remaining: 0 (0 B) Cache Used%: 100.00% Cache Remaining%: 0.00% Xceivers: 10

Last contact: Wed Jul 20 04:55:01 GMT 2022 Last Block Report: Wed Jul 20 02:36:21 GMT 2022 Num of Blocks: 149

root@h01:/usr/local/hadoop/bin#

访问宿主机的 8088 与 9870 端口就可以看到监控信息了

	H	Hadoop c	werview Data	anodes Data <u>nor</u>	le Volume Failures	Snapshot Sta	rtup Progress	Utilities 🖌							
	(Overvi	ew 'h01:	:9000' (~ act	ve)										
		Started:		Sun Jul 17	16:47:56 +0800 2022										
		Version:		3.3.3, rd375	86cbda38c338d9fe481	addda5a05fb516l	71								
		Compiled:		Tue May 10	00:36:00 +0800 2022 k	y stevel from bra	nch-3.3.3								
		Cluster ID:		CID-0bb5b	l9c-bc66-4197-9a4a-3d	71274d6bdb									
		Block Pool ID:	:	BP-157550	8096-172.18.0.2-165804	15878492									
		Summ	ary												
	s	Security is off.													
	S	Safemode is off.													
	3	83 files and dire	ectories, 309 bloc	ks (309 replicated b	locks, 0 erasure coded I	block groups) = 6	92 total filesystem	n object(s).							
	r N	lon Heap Memo	ory used 74.75 M	B of 76.67 MB Com	nory. Max Heap Memor	y is 1.73 GB. y. Max Non Heap	Memory is <unbo< td=""><td>ounded>.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></unbo<>	ounded>.							
		Configured Ca	apacity:				84.5	57 GB							
		Configured Re	emote Capacity:				0 B								
		DFS Used:					8.11	L MB (0.01%)							
		Non DFS Used	d:				55.3	36 GB							
		DFS Remainin	ıg:				22.4	44 GB (26.53%)							
		Block Pool Us	ed:				8.11	L MB (0.01%)							
		DataNodes us	ages% (Min/Med	dian/Max/stdDev):			0.01	1% / 0.01% / 0.029	6/0.00%						
							5.0	ecommissioned (), In Maintenar	ce: 0)					
1 封未读)— Evoluti (型 Fin ecluster × + ▲ 不安全 192.168.1.20 ouTube ♀ 地图 嗪 翻	nalShell 3.9.5.7 :8088/cluster/no 译 💽 在线节拍	Live Nodes	oogle-chrome 操作系统(双语)	 Typore C (26条消息) 	的下载及MarkDow 至 nanja C (27条消息)	基于 Docker 构 Manja 🧲 (27	☆ Hadoop ● 1	Wine ♀ manjaro5pyth	· ·o 课程:	P心·辽宁省	Q Issues · chxu	an/v 💽 (275	▲ <mark>■</mark> 配 : S消息) 从pyt	\$ Ø □ 囲 ♥) > ☆ ま IJ € (27条消息) virr) ≡ 20 ~ T ∎ 1 m12
1 封未該) — Evoluti. 一 Pr icluster × + ▲ 不安全 192.168.1.20 iouTube ♥ 地面 ■ 翻 こここの「」	nalShell 3.9.5.7 :8088/cluster/no 译 2 在线节拍	Live Nodes Godes 쫋 Mus 한	oogle-chrome 操作系统(双语)	Typore C (26条消息)	的下载及MarkDow 1 nanja [] (27条消息)	基于 Docker 构 Manja C (27 Nod	Hadoop (す) 条消息) vim 配 es of th	wine ৩৻ manjaro5pyth e cluste	io 课程* er	₽心-辽宁省	O Issues - chxu	an/v C (27\$	● ■ 配 : \$消息) 从pyt	◆ & □ 田 ●) > ☆ ★ 弐 C (27条消息) virr) 三 20
1 封決法) — Evoluti. 『 19 1 cluster × + ▲ 不安全 192.168.1.20 00010be ♥ 総理 副 このののののののののののののののののののののののののののののののののののの	nalShell 3.9.5.7 :8088/cluster/nd 译 1 在线节拍 C trics mitted Apps o des Metrics	Live Nodes Godes 왕 Mus 한 Pending A 0	bogle-chrome 操作系统(双语) pps Running	Typord (25条消息) Apps Completed 1	的下部就MarkDow manja Containers Runnin 0	基于 Docker 构 Manja ⓒ (27 Nod g Used <memory:0< td=""><td>E Hadoop ⑥ 条准息) vim 配 es of th Resources B, vCores:0></td><td>Mine ای manjaro5pyth e cluste Total <memory:40 g<="" td=""><td>io 课程 C Resources B, vCores:402</td><td>P心辽宁省… R <mer< td=""><td>O Issues - chxu eserved Resource rory:0 B, vCores:</td><td>es Phys 0> 94</td><td>▲ 【】 @{ ; \$消息) 从pyt</td><td> </td><td>) 三 20 ~ 「 □ Logg al VCore</td></mer<></td></memory:40></td></memory:0<>	E Hadoop ⑥ 条准息) vim 配 es of th Resources B, vCores:0>	Mine ای manjaro5pyth e cluste Total <memory:40 g<="" td=""><td>io 课程 C Resources B, vCores:402</td><td>P心辽宁省… R <mer< td=""><td>O Issues - chxu eserved Resource rory:0 B, vCores:</td><td>es Phys 0> 94</td><td>▲ 【】 @{ ; \$消息) 从pyt</td><td> </td><td>) 三 20 ~ 「 □ Logg al VCore</td></mer<></td></memory:40>	io 课程 C Resources B, vCores:402	P心辽宁省… R <mer< td=""><td>O Issues - chxu eserved Resource rory:0 B, vCores:</td><td>es Phys 0> 94</td><td>▲ 【】 @{ ; \$消息) 从pyt</td><td> </td><td>) 三 20 ~ 「 □ Logg al VCore</td></mer<>	O Issues - chxu eserved Resource rory:0 B, vCores:	es Phys 0> 94	▲ 【】 @{ ; \$消息) 从pyt) 三 20 ~ 「 □ Logg al VCore
1 計表記) — Crobute 一 Fr relater × + 本 不安全 192.168.1.20 courtube 9 地面 計算 Cluster No. 1 Cluster No. 2 5	nalSheil 3.9.9.7 :8088/cluster/nu 译 图 在线节拍 D trics 0 des Metrics	Live Nodes	oogle-chrome 操作系统(双语) pps Running Accommissioning	Typora … C (26条消息) Apps Completed 1 Nodes	合下在認知体 RDow. T manja. (2)(27条元息) Containers Runnin 0 Decomm 0	Manja @ (27 Nod g Used <memory:0< td=""><td>E Hadoop ⑧ 条消息) vim 配 es of th Resources B, vCores:0></td><td>Mine العام (Mine) (Mi</td><td>io 课程 PT Resources B, vCores:402</td><td>やいび字省… やひ辽字省… R ・ <merr< td=""><td>C Issues - chxu eserved Resourc nory:0 B, vCores: es</td><td>es Phys Rebooted</td><td>● 二</td><td> </td><td>D = 20</td></merr<></td></memory:0<>	E Hadoop ⑧ 条消息) vim 配 es of th Resources B, vCores:0>	Mine العام (Mine) (Mi	io 课程 PT Resources B, vCores:402	やいび字省… やひ辽字省… R ・ <merr< td=""><td>C Issues - chxu eserved Resourc nory:0 B, vCores: es</td><td>es Phys Rebooted</td><td>● 二</td><td> </td><td>D = 20</td></merr<>	C Issues - chxu eserved Resourc nory:0 B, vCores: es	es Phys Rebooted	● 二	 	D = 20
1 計畫書) - Crobitit - 一 rf rduster × + 本 不安全 192.168.1.20 wurube や 地面 単語 「Cluster No. 5 Cluster No. 5 Schedduer F 5 Schedduer H	nalShell 3.9.5.7 :8088/cluster/nn 译 图 在线节拍 ① ① clust mitted Apps 0 des Metrics re Nodes Metrics wether: Thre	Live Nodes	oogle chrome 建作系统(双语) pps Running Scheddian Da	Z Typore … C (26条消息) Apps Completed 1 Nodes	的下記23M#KDow. ① nanja. ② (27年3月巻) Containers Runnin の Decomm Q	当于 Docker 約 Manja € (27 Nod g Usec <memory:0 issioned Nodes um Allocation</memory:0 	は 健 Hadoop → 会 小 他	بلاست بر معالی می	io 课程 PT Resources B, vCores:40:2	P心辽宁省… R <mer nhealthy Nod</mer 	C Issues - chxu esserved Resource esserved Resource es g Mavimum C	es Physicalin Rebooted) して 注 部語 別 从pyt ical Mem Use Nodes	◆ ∂ □ Ⅲ ◆ ☆ ∂ □ Ⅲ ◆ ☆ ★ ゴ/ ○ (27条消息) virr 6 % Physica 0 Shutde 0 5 6 5<) = 20
1 11年初一 Evolution の frie e cluster × ・ 本 不安全 192-168-1.20 Wurking や 地理 単 前 ここのこのの には、 たいませた Me ないませた Activ 5 5 5 5 5 5 5 5 5 5 5 5 5	atshell 3.9.5.7 :8088/cluster/nd 译 @ 在线节拍 ① ① /////////////////////////////////	Live Nodes	oogle chrome Iarts #ste (3218) upps Running becommissioning Scheduling Re -mb (unit=M), vc	Typera Typera Typera (20条消息) Apps Completed Nodes source Type orres]	合下部33MarkDow	Manja	また 使 Hádoop … 合 奈清息) vim 起- es of th Resources 泉, vCores O> の 、 、 、 、 、 、 、 、 、 、 、 、 、	Wine	io 课程 Casources B, vCores:40-2 Q loccation 4-2	やひ辽宁省 R - <men nhealthy Nod</men 	Issues - drau eserved Resource eserved Resource ese Q Maximum Cl	es Physicalio Rebooted	오는 말 (전) : *개왕) 从pyt ical Mem Use) = 20
1 11日本 2 11日本 2 11日本 本 不安全 1 192.168.1.20 Wurfube 9 松弛 単語 1 192.068.1.20 Wurfube 9 松弛 単語 1 192.068.1.20 Cluster No. 1 200 1 200	Halshell 3.9.5.7 Halshell 3.	Live Nodes	oogle chrome Betrs Ket (Xila) pps Running becommissioning Scheduling Re mb (unit=Mi), vc Address	Typercompleted Apps Completed Nodes Source Type Nodes Nodes	وَ کَلَیْکَ اللَّهُ ال Containers Runnin O Decomm Q Last health- update	X∓ Docker (b) Manja € (27 Nod g Usec g Usec um Allocation Cores:1>	(Resources 泉,水のい) 泉, vores. ひ の (Containers	Wine Wine A manjaro-5pyth e cluste commony.40 G Lost Nodes Maximum All Maximum All Allocation Tags	io 课程 PT Resources B, vCores:402 U Q U Ccation 42	R R R R R R R R R R R R R R R R R R R	C Issues - chxu eserved Resources onry:0 B, vCores: 9 Maximum Cl Merm Used W.	es Physicatio Rebooted VCores Used	。 こ 本語的 Apyt ical Mem Use Nodes a Priority VCorers Avail) = 20
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Halshell 3.9.5.7 #3088/cluster/in #3 查 在线节始 #3 查 在线节始 #4 #4 #4 #4 #4 #4 #4 #4 #4 #4	Live Nodes	oogle-chrome 時代系統(双语) pps Running becommissioning Scheduling Re- mit (unit=N), vc Node Address h02:38365	Typer Typer Typer Apps Completed A Nodes Source Type Node HTTP Address hozesu2	今下彰33/MarkDow 『 nanja. ② (27条決意) Containers Runnin の Decomm の したのいたい く物理的などの になった。 の の したのいたい の の したのいたい の したのの したの	Manja C (27 Nod sissioned Nodes sissioned Nodes um Allocation Cores 1> Health- report	(日本) (日	Wine VA manjaroEpyth e cluster Total Total Anony 40 G Lost Nodes Maximum Al Maximum Al Allocation Tags	io 课程 Resources B, vCores:40> 0 0 ccation 4> 0 0 B	R 小山市省一 R 小山市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市市	Ω Issues - chau eserved Resource ony 0 B, vCores: es Q Maximum Cl Phys Used % 94	es Physical Control of	。 こ 二 二 二 二 二 二 二 二 二 二 二 二 二	A B B B B B B B B B B B B B B B B B) ≡ 20
1 まままま) - Krokket - 『 fr eduster × ・ 本 不安全 192.168.120 varTube ・ 松田 - 日 に に は 、 、 、 、 、 、 、 、 、 、 、 、 、	AdShell 3.9.5.7	Live Nodes	oogle-chrome 諸性系統(双语) pps Running Scheduling Re- mb (uniE-M), vc Node Nod2:38365 h04:38589	Typore Typore (26%782)* Apps Completed 1 Nodes source Type ores) Node HTTP Address ho2.8042 ho4.8042	会下を記込MarkCow. 2 (27条河急) Containers Runnin 0 Decomm 2 Decomm 2 Minim <nemory.1024, Web Jul 20 002022 (2000) 2022</nemory.1024, 	Manja C (27 Nod usecond construction issioned Nodes num Allocation Cores:1> Health- report	Kitesources Si清島) vim 配 ess of th Resources R, vCores ン- の Containers 0	Mine Vin manjaro5pyth e cluste Total cremory.40 Lost Nodes Maximum All Maximum All Maxim	ao 课程 Cr Resources B, vCores.400 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	R 中心辺字音… R <mem ahealthy Nod の</mem 	C Issues - chau eserved Resource nory 0 B, vCores: es 0 Maximum Cl Maximum Cl Phys Mem 94 94	es Physication Rebooted Uster Application VCores 0 0	。 こ こ こ こ こ こ こ こ こ こ こ こ こ) = 20
1 引き3) - Croduct - 『 rf ectuster × + 本 不安全 192.168.120 courtube 9 地面 単語 Cluster Not 1 Cluster Not 2 Scheddurf 1 Courter Not 2 Scheddurf 1 Courter Not 2 Scheddurf 1 1 1 1 1 1 1 1 1 1 1 1 1	nalsheil 3.9 5.7 ROBB/cluster/Ind ROBB/cluster/Ind ROBB/cluster/Ind ROBB/cluster/Ind ROBB/cluster/Ind ROBB/cluster/Ind Robbit ROBB/cluster/Ind ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind Robbit ROBB/cluster/Ind RO	Live Nodes	eogle chrome Bertr Siek (3218) pps Running becommissioning Scheduling Re- mb (unit=N6), vc Node Address ho2:38365 ho3:40897	Typer Typer Typer Appa Completed Nodes Source Type Node HTTP Address ho2.8042 ho3.8042	会下を見込めたKDow で manja (2) (27条洗魚) Containers Runnin 0 Decomm 0 Decomm 4 Containers Runnin 0 Decomm 4 Containers Runnin 0 Decomm 4 Containers Runnin 0 Decomm 4 Containers Runnin 0 Decomm 4 Containers Runnin 0 Decomm 5 Containers Runnin 0 Decomm 4 Containers Runnin 0 Decomm 5 Containers Runnin 0 Decomm 5 Containers Runnin 0 Decomm 5 Containers Runnin 0 Decomm 5 Containers Runnin 0 Containers Runnin 0 Decomm 5 Containers Runnin 0 Containers Runnin 0 Contai	Manja C (27 Nod sissioned Nodes aum Allocation Cores 1> Health- report	、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、	و cluster e cluster آتاما درستان وی درمانه المعامر بالی درمانه المعامر بالی درمانه المعامر بالی درمانه المعامر معامر المعامر المعامر المعامر	io 课程 Cesources B, vCores:400 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	R R (men B CB B CB B CB	Summer in the second	es Physication Rebooted Useer Application 0 0 0	الله المعالية ال المعالية المعالية المعالية المعالية المعالية المع المعالية المعالية الم	◆ ♪ □ □ ● ◆ ♪ ☆ ♪ ■ ■ ◆ ☆ ♪ ■ G (7意讯息) viri G (7意讯息) viri 0 0 0 Scheb 0 Schud 0 Schud 0 Schud 0 Schud 5 5 5 5) = 20
1 当我歌) - Drobinu - ⑦ fr eduster × ・ 本 不安全 192.168.120 rouTube ◆ 地感 章 田 Cluster Nor Cluster Nor Cluster Nor Apps Subt 1 Cluster Nor Cluster	NaiSheil 3.9.9.7 ROBB/cluster/no ROBB/cluster/no ROBB/cluster/no Robe Metrics Apps des Metrics Apps des Metrics Apps des Metrics Apps des Metrics Metrics Apps Apps des Metrics App	Live Nodes	ecopie chrome Betr Kete (2018) pps Running Scheduling Re- mb (unit=M), vc Node h02:38305 h04:38989 h03:40897 h05:33487	Typoco Typoco Completed Completed Nodes Source Type Node HTTP Address h02.8042 h03.8042	(今下私33:Mar kCow て のすいない。 のまれ」。 このれないであった。 のでのれないであった。 のでのようない。 のでのようない。 のでのようない。 のでので、1024、い のでのでので、1024、い のでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでので	Manja C (27 Nod g Usec «memory:0 issioned Nodes um Allocation Cores:1> Health- report	Resources の の Containers 0 0 0 0 0 0	Mine Mine Mine Spyth A manjaroSpyth Cost Notes Maximum All Maximum All Maxim	(0	R R criter Inhealthy Nod 8 GB 8 GB 8 GB 8 GB 8 GB	Issues - chau eserved Resource onory 0 B, vCores: o Maximum Cl Maximum Cl Phys Maximum Cl Phys Maximum Cl Phys P4 94 94 94 94 94 94 94 94 94 94 94 94 94	es Physicatio Rebooted VCores 0 0 0 0	الله الله الله الله الله الله الله الل	Image: Second) = 20 v mB2 Logy lown Nc eduler I s 6
1 11日日 - 1000000 - 日 「「 coluster × + 本 不安全 1921-108-1.20 wurube 9 松照 日 日 Cluster No. 1 Cluster No. 1 Cluster No. 5 Schedderdiser Schedwick Stow 20 0 	Alshell 3.9.5.7	Live Nodes	هوراف دارسته هراف	Typer Typer Type Appa Completed Appa Completed Nodes Source Type Node HTTP Node HTTP Node HTTP Node S042 h04.8042 h05.8042 h05.8042 h03.8042 h03.8	(今下登込が4k00w. で (27条洗魚) Containers Runnin 0 Decomm 0 Decomm 0 Last health- update 0 000022 +0000 2022 Wed Jul 20 05:002 2 +0000 2022 Wed Jul 20 05:002 + 0000 2022	Manja C (27 Nod g Usec sistoned Nodes arm Allocation Cores 1> Health- report	Hadoop) で Addeop) で Addeop) で	Vine Vine	Resources Resources 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	R R Creation R Creation R R Creation R R R R R R R R R R R R R R R R R R R	C Issues - chuu esserved Resource onr.0 B, vCores: es Q Maximum Cl Maximum Cl Maximum Cl Maximum Cl Phys Meen Used Va 94 94 94 94 94 94	es Physical Rebooted uster Applicatio VCorres Used 0 0 0 0 0 0	A The second secon) = 20
1 引きま) - Evolute - 『 fr actuster × + 本 不安全 192.163.120 rouTube 学 地面 単語 Cluster Not Active Scheduler 1 Scheduler 1	alshell 3.9.5.7 BOBB/cluster/not BOBB/cluster/not BOBB/cluster/not trics Cluster/not Clus	Live Nodes	bogle-chrome B#f7.#68_0XLB pps Running becommissioning Scheduling Re- mb (unit=M0), vc Node h04:38989 h03:40897 h05:33497 h01:34379	Typece Typece Typece Apps Completed A Nodes Sance Type Nodes		Manja C (27 Nod g Used «memory:0 issioned Nodes um Allocation (Cores:1>	・ 「 Hadooy) 。 会 消息) vim 紀- ess of th Resources り、 Containers 0 0 0 0 0 0	Mine المعايمة المحمد المحمد محمد المحمد المحم الم محمد المحمد المحم المحمد المح المحمد المحمد المحم المحمد المحمد المحم المحمد ا	0 WHY Products V 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Rebuild of the second s	Bases - draw eserved Resource O By vCores: O Maximum CI Phys Mem Used % 94 94 94 94 94	es Physe Rebooted Uster Applicatio	الله الله الله الله الله الله الله الله	Image: Control of the second	E 200 E
1 引きま) - Evolute - 『 fr actuster × + 本 不安全 192168.120 courtube 学 地面 単語 Cluster Not S Cluster Not Capacity Sc Schedderl Labels Show: 00 1 to Showing 1 to	alshell 3.9.5.7 S088X/Luster/not S088X/Luster/not trics Comment Comme	Live Nodes Codes C	cogle-chrome litefrs.#de (2018) pps Running pps Running scheduling Re -mb (unit=Mb), vc Nodes Address ho4:36969 ho3:40697 ho1:34379	Typer T	(今下記33/Mar kDow て ののしていたいでは、 のでしたいでは、 のでいでは、 のでいでは、 のでしたいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでは、 のでいでいでいでいでは、 のでいでは、 のでいでいでいでいでいでいでいでいでいでいでいでいでいでいでいでいでいでいでい	Manja C (27 Nod g Used «memory:0 issioned Nodes um Allocation (Cores:1>	(1) (1) (1) (1) (1) (1) (1) (1) (1)	الالله المعالمة المعا المعالمة المعالمة الم المعالمة المعالمة الم المعالمة المعالمة الم المعالمة المعالمة المع المعالمة المعالمة المعالمة معالمة المعالمة المعالم معالمة معالمة مع معالمة معالمة م معالمة معالمة مع معالم معالمة معال	0. WHY Pr Pr Resources 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Rebuild of the second s	Baues - draw eserved Resource O A A A A A A A	es Physo Rebooted Uster Applicatio 0 0 0 0 0	الله الله الله الله الله الله الله الله	A A	E 20 E 20 T IN THE SECOND SECON

至此, Hadoop 集群已经构建好了

2.2.5 运行内置WordCount例子

把 license 作为需要统计的文件

```
root@h01:/usr/local/hadoop# cat LICENSE.txt > file1.txt
root@h01:/usr/local/hadoop# ls
```

上传 file1.txt 文件到 HDFS 中

```
root@h01:/usr/local/hadoop/bin# ./hadoop fs -put ../file1.txt /input
root@h01:/usr/local/hadoop/bin#
```

查看 HDFS 中 input 文件夹里的内容

```
root@h01:/usr/local/hadoop/bin# ./hadoop fs -ls /input
Found 1 items
-rw-r--r- 2 root supergroup 15217 2022-07-17 08:50 /input/file1.txt
root@h01:/usr/local/hadoop/bin#
```

运行wordcount 例子程序

root@h01:/usr/local/hadoop/bin# ./hadoop jar ../share/hadoop/mapreduce/hadoopmapreduce-examples-3.3.3.jar wordcount /input /output

输出如下:

```
root@h01:/usr/local/hadoop/bin# ./hadoop jar ../share/hadoop/mapreduce/hadoop-
mapreduce-examples-3.3.3.jar wordcount /input /output
2022-07-20 05:12:38,394 INFO client.DefaultNoHARMFailoverProxyProvider:
Connecting to ResourceManager at h01/172.18.0.2:8032
2022-07-20 05:12:38,816 INFO mapreduce.JobResourceUploader: Disabling Erasure
Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1658047711391_0002
2022-07-20 05:12:39,076 INFO input.FileInputFormat: Total input files to process
: 1
2022-07-20 05:12:39,198 INFO mapreduce.JobSubmitter: number of splits:1
2022-07-20 05:12:39,399 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1658047711391_0002
2022-07-20 05:12:39,399 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-07-20 05:12:39,674 INFO conf.Configuration: resource-types.xml not found
2022-07-20 05:12:39,674 INFO resource.ResourceUtils: Unable to find 'resource-
types.xml'.
2022-07-20 05:12:39,836 INFO impl.YarnClientImpl: Submitted application
application_1658047711391_0002
2022-07-20 05:12:39,880 INFO mapreduce.Job: The url to track the job:
http://h01:8088/proxy/application_1658047711391_0002/
2022-07-20 05:12:39,882 INFO mapreduce.Job: Running job: job_1658047711391_0002
2022-07-20 05:12:49,171 INFO mapreduce.Job: Job job_1658047711391_0002 running in
uber mode : false
2022-07-20 05:12:49,174 INFO mapreduce.Job: map 0% reduce 0%
2022-07-20 05:12:54,285 INFO mapreduce.Job: map 100% reduce 0%
2022-07-20 05:13:01,356 INFO mapreduce.Job: map 100% reduce 100%
2022-07-20 05:13:02,391 INFO mapreduce.Job: Job job_1658047711391_0002 completed
successfully
2022-07-20 05:13:02,524 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=12507
                FILE: Number of bytes written=577413
                FILE: Number of read operations=0
```

```
FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=15313
                HDFS: Number of bytes written=9894
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3141
                Total time spent by all reduces in occupied slots (ms)=3811
                Total time spent by all map tasks (ms)=3141
                Total time spent by all reduce tasks (ms)=3811
                Total vcore-milliseconds taken by all map tasks=3141
                Total vcore-milliseconds taken by all reduce tasks=3811
                Total megabyte-milliseconds taken by all map tasks=3216384
                Total megabyte-milliseconds taken by all reduce tasks=3902464
       Map-Reduce Framework
                Map input records=270
                Map output records=1672
                Map output bytes=20756
                Map output materialized bytes=12507
                Input split bytes=96
                Combine input records=1672
                Combine output records=657
                Reduce input groups=657
                Reduce shuffle bytes=12507
                Reduce input records=657
                Reduce output records=657
                Spilled Records=1314
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=126
                CPU time spent (ms)=1110
                Physical memory (bytes) snapshot=474148864
                Virtual memory (bytes) snapshot=5063700480
                Total committed heap usage (bytes)=450887680
                Peak Map Physical memory (bytes)=288309248
                Peak Map Virtual memory (bytes)=2528395264
                Peak Reduce Physical memory (bytes)=185839616
                Peak Reduce Virtual memory (bytes)=2535305216
       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=15217
       File Output Format Counters
                Bytes Written=9894
root@h01:/usr/local/hadoop/bin#
```

```
root@h01:/usr/local/hadoop/bin# ./hadoop fs -ls /output
Found 2 items
-rw-r--r- 2 root supergroup 0 2022-07-20 05:13 /output/_SUCCESS
-rw-r--r- 2 root supergroup 9894 2022-07-20 05:13 /output/part-r-00000
root@h01:/usr/local/hadoop/bin#
```

查看 part-r-00000 文件的内容

root@h01:/usr/local/hadoop/bin# ./hadoop fs -cat /output/part-r-00000

至此, hadoop部分已经结束

2.3 安装 Hbase

在 Hadoop 集群的基础上安装 Hbase

下载 Hbase 3.0.0

```
root@h01:~# wget https://mirrors.tuna.tsinghua.edu.cn/apache/hbase/3.0.0-alpha-
3/hbase-3.0.0-alpha-3-bin.tar.gz
```

解压到 /usr/local 目录下面

root@h01:~# tar -zxvf hbase-3.0.0-bin.tar.gz -C /usr/local/

修改 /etc/profile 环境变量文件,添加 Hbase 的环境变量,追加下述代码

export HBASE_HOME=/usr/local/hbase-3.0.0
export PATH=\$PATH:\$HBASE_HOME/bin

使环境变量配置文件生效

root@h01:/usr/local# source /etc/profile root@h01:/usr/local#

使用 ssh h02 可进入h02容器,修改profile文件如上。依次修改h03、h04、h05

即是每个容器都要在 /etc/profile 文件后追加那两行环境变量

在目录 /usr/local/hbase-3.0.0/conf 修改配置

修改 hbase-env.sh,追加

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HBASE_MANAGES_ZK=true

修改 hbase-site.xml 为

<configuration>
 <property>
 <name>hbase.rootdir</name>

```
<value>hdfs://h01:9000/hbase</value>
        </property>
        <property>
                <name>hbase.cluster.distributed</name>
                <value>true</value>
        </property>
        <property>
                <name>hbase.master</name>
                <value>h01:60000</value>
        </property>
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>h01,h02,h03,h04,h05</value>
        </property>
        <property>
                <name>hbase.zookeeper.property.dataDir</name>
                <value>/home/hadoop/zoodata</value>
        </property>
</configuration>
```

修改 regionservers 文件为

h01 h02 h03 h04 h05

使用 scp 命令将配置好的 Hbase 复制到其他 4 个容器中

```
root@h01:~# scp -r /usr/local/hbase-3.0.0 root@h02:/usr/local/
root@h01:~# scp -r /usr/local/hbase-3.0.0 root@h03:/usr/local/
root@h01:~# scp -r /usr/local/hbase-3.0.0 root@h04:/usr/local/
root@h01:~# scp -r /usr/local/hbase-3.0.0 root@h05:/usr/local/
```

启动 Hbase

root@h01:/usr/local/hbase-3.0.0/bin# ./start-hbase.sh h04: running zookeeper, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase-rootzookeeper-h04.out h02: running zookeeper, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase-rootzookeeper-h02.out h03: running zookeeper, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase-rootzookeeper-h03.out h05: running zookeeper, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase-rootzookeeper-h05.out h01: running zookeeper, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase-rootzookeeper-h01.out running master, logging to /usr/local/hbase-3.0.0/bin/../logs/hbase--masterh01.out h05: running regionserver, logging to /usr/local/hbase-3.0.0/bin/../logs/hbaseroot-regionserver-h05.out h01: running regionserver, logging to /usr/local/hbase-3.0.0/bin/../logs/hbaseroot-regionserver-h01.out h04: running regionserver, logging to /usr/local/hbase-3.0.0/bin/../logs/hbaseroot-regionserver-h04.out h03: running regionserver, logging to /usr/local/hbase-3.0.0/bin/../logs/hbaseroot-regionserver-h03.out h02: running regionserver, logging to /usr/local/hbase-3.0.0/bin/../logs/hbaseroot-regionserver-h02.out root@h01:/usr/local/hbase-3.0.0/bin#

打开 Hbase 的 shell

```
root@h01:/usr/local/hbase-3.0.0/bin# ./hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/book.html#shell
Version 3.0.0-alpha-3, rb3657484850f9fa9679f2186bf53e7df768f21c7, Wed Jun 15
07:56:54 UTC 2022
Took 0.0017 seconds
```

hbase:001:0>

hbase测试

创建表member

```
hbase:006:0> create 'member','id','address','info'
Created table member
Took 0.6838 seconds
=> Hbase::Table - member
hbase:007:0>
```

添加数据,并查看表中数据

hbase:007:0> put 'member', 'debugo','id','11' Took 0.1258 seconds

hbase:008:0> put 'member', 'debugo', 'info:age', '27'

```
Took 0.0108 seconds
hbase:009:0> count 'member'
1 row(s)
Took 0.0499 seconds
=> 1
hbase:010:0> scan 'member'
ROW
                                          COLUMN+CELL
                                          column=id:, timestamp=2022-07-
debugo
20T05:37:58.720, value=11
debugo
                                          column=info:age, timestamp=2022-07-
20T05:38:11.302, value=27
1 row(s)
Took 0.0384 seconds
hbase:011:0>
```

2.4 安装 Spark

在 Hadoop 的基础上安装 Spark

下载 Spark 3.3.0

```
root@h01:~# wget https://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-
3.3.0/spark-3.3.0-bin-hadoop3.tgz
```

解压到 /usr/local 目录下面

root@h01:~# tar -zxvf spark-3.3.0-bin-hadoop3.tgz -C /usr/local/

修改文件夹的名字

```
root@h01:~# cd /usr/local/
root@h01:/usr/local# mv spark-3.3.0-bin-hadoop3 spark-3.3.0
```

修改 /etc/profile 环境变量文件,添加 Hbase 的环境变量,追加下述代码

export SPARK_HOME=/usr/local/spark-3.3.0
export PATH=\$PATH:\$SPARK_HOME/bin

使环境变量配置文件生效

```
root@h01:/usr/local# source /etc/profile
root@h01:/usr/local#
```

使用 ssh h02 可进入其他四个容器, 依次修改。

即是每个容器都要在 /etc/profile 文件后追加那两行环境变量

在目录 /usr/local/spark-3.3.0/conf 修改配置

修改文件名

root@h01:/usr/local/spark-3.3.0/conf# mv spark-env.sh.template spark-env.sh

root@h01:/usr/local/spark-3.3.0/conf#

修改 spark-env.sh, 追加

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export SCALA_HOME=/usr/share/scala
export SPARK_MASTER_HOST=h01
export SPARK_MASTER_IP=h01
export SPARK_WORKER_MEMORY=4g
```

修改文件名

```
root@h01:/usr/local/spark-3.3.0/conf# mv slaves.template slaves
root@h01:/usr/local/spark-3.3.0/conf#
```

修改 slaves 如下

h01 h02 h03 h04 h05

使用 scp 命令将配置好的 Hbase 复制到其他 4 个容器中

```
root@h01:/usr/local# scp -r /usr/local/spark-3.3.0 root@h02:/usr/local/
root@h01:/usr/local# scp -r /usr/local/spark-3.3.0 root@h03:/usr/local/
root@h01:/usr/local# scp -r /usr/local/spark-3.3.0 root@h04:/usr/local/
root@h01:/usr/local# scp -r /usr/local/spark-3.3.0 root@h05:/usr/local/
```

启动 Spark

```
root@h01:/usr/local/spark-3.3.0/sbin# ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark-
3.3.0/logs/spark--org.apache.spark.deploy.master.Master-1-h01.out
h03: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-h03.out
h02: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-h02.out
h04: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-h04.out
h05: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.loging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.logging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.loging to /usr/local/spark-
3.3.0/logs/spark-root-org.apache.spark.deploy.worker.Worker.loot.
```

3.1 HDFS 重格式化问题

参考 https://blog.csdn.net/gis_101/article/details/52821946

- 重新格式化意味着集群的数据会被全部删除,格式化前需考虑数据备份或转移问题;
- 先删除主节点(即namenode节点),Hadoop的临时存储目录tmp、namenode存储永久性元数 据目录dfs/name、Hadoop系统日志文件目录log中的内容(注意是删除目录下的内容不是目录);
- 删除所有数据节点(即datanode节点), Hadoop的临时存储目录tmp、namenode存储永久性元数 据目录dfs/name、Hadoop系统日志文件目录log 中的内容;
- 格式化一个新的分布式文件系统:

root@h01:/usr/local/hadoop/bin# ./hadoop namenode -format

注意事项:

- Hadoop的临时存储目录tmp(即core-site.xml配置文件中的hadoop.tmp.dir属性,默认值 是/tmp/hadoop-\${user.name}),如果没有配置hadoop.tmp.dir属性,那么hadoop格式化时将 会在/tmp目录下创建一个目录,例如在cloud用户下安装配置hadoop,那么Hadoop的临时存储目 录就位于/tmp/hadoop-cloud目录下
- Hadoop的namenode元数据目录(即hdfs-site.xml配置文件中的dfs.namenode.name.dir属性, 默认值是\${hadoop.tmp.dir}/dfs/name),同样如果没有配置该属性,那么hadoop在格式化时将 自行创建。必须注意的是在格式化前必须清楚所有子节点(即DataNode节点)dfs/name下的内 容,否则在启动hadoop时子节点的守护进程会启动失败。这是由于,每一次format主节点 namenode,dfs/name/current目录下的VERSION文件会产生新的clusterID、namespaceID。但 是如果子节点的dfs/name/current仍存在,hadoop格式化时就不会重建该目录,因此形成子节点 的clusterID、namespaceID与主节点(即namenode节点)的clusterID、namespaceID不一致。 最终导致hadoop启动失败。