Docker下安装Hadoop集群

1. 安装Centos容器

1.1 下载Centos镜像

```
[root@kmaster ~]# docker pull centos:centos7
```

查看集群镜像：

```
[root@kmaster ~]# docker images
REPOSITORY                                                          TAG                 IMAGE
ID             CREATED             SIZE
centos                                                              centos7
eeb6ee3f44bd        7 months ago        204MB
centos                                                              latest
5d0da3dc9764        7 months ago        231MB
[root@kmaster ~]#
```

1.2 启动Centos镜像

```
[root@kmaster ~]# docker run -d --name master --privileged=true centos:centos7 /usr/sbin/init
85de83414bfe5bbf278717e91ebe96276c4580db7b67465bcc8690c88a337095
[root@kmaster ~]#
```

查看容器进程：

```
[root@kmaster ~]# docker ps
CONTAINER ID        IMAGE                                               COMMAND
   CREATED             STATUS              PORTS                    NAMES
1fccc346a422        centos:centos7                                        "/bin/bash"
     10 seconds ago       Up 9 seconds                                  master

```

1.3 进入容器Master

```
[root@kmaster ~]# docker exec -it master /bin/bash
```

2. 调整Centos容器

2.1 初始化容器环境

```
[root@39bc9eb3afa1 /]# yum check-update -y
[root@39bc9eb3afa1 /]# yum update -y
[root@39bc9eb3afa1 /]# yum install initscripts screen wget -y
```

2.2 给容器安装基础服务

1. 按装ssh服务和网络必须软件:

```
[root@1fccc346a422 /]# yum install net-tools.x86_64 -y
[root@1fccc346a422 /]# yum install openssh-server -y
```

2. 安装完后重启SSH服务:

```
[root@85de83414bfe /]# systemctl restart sshd
[root@85de83414bfe /]#
```

3. 安装passwd软件（用于设置centos用户密码，便于用Xshell连接):

```
[root@85de83414bfe /]# yum install passwd -y
```

设置root用户密码：

```
[root@85de83414bfe /]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@85de83414bfe /]#
```

4. 安装scp远程拷贝:

```
[root@36aea7d07044 ~]# yum -y install openssh-clients
```

5. 安装防火墙服务：

```
[root@85de83414bfe /]# yum -y install firewalld
```

6. 安装which：

```
[root@85de83414bfe /]# yum -y install which
```

3. 保存Centos容器

1. 查看容器id

```
[root@kmaster ~]# docker ps
CONTAINER ID        IMAGE                                               COMMAND
   CREATED             STATUS              PORTS                    NAMES
85de83414bfe        centos:centos7                                         "/usr/sbin/init"
    17 hours ago         Up 17 hours                                 master
```

2. 将容器id 提交成为镜像：

```
[root@kmaster ~]# docker commit 85de83414bfe centos7:v1
```

3. 将容器镜像推送到本地仓库:

```
[root@kmaster ~]# docker push 85de83414bfe centos7:v1
```

4. 使用保存的镜像部署hadoop集群

4.1 生成centos容器并将容器组成集群

4.1.1 使用保存的镜像生成容器

1. 生成容器slave1：

```
[root@kmaster ~]# docker run -d --name slave1 --privileged=true centos7:v1 /usr/sbin/init
85de83414bfe5bbf278717e91ebe96276c4580db7b67465bcc8690c88a337095
[root@kmaster ~]#
```

2. 生成容器slave2：

```
[root@kmaster ~]# docker run -d --name slave2 --privileged=true centos7:v1 /usr/sbin/init
85de83414bfe5bbf278717e91ebe96276c4580db7b67465bcc8690c88a337095
[root@kmaster ~]#
```

4.1.2 修改容器内的主机名，方便后续辨认:

1. 进入master中修改主机名:

```
[root@kmaster ~]# docker exec -it master /bin/bash
[root@85de83414bfe /]# hostnamectl set-hostname master
[root@85de83414bfe /]# bash
[root@master /]#
```

2. 在slave1中:

```
[root@kmaster ~]# docker exec -it slave1 /bin/bash
[root@85de83414bfe /]# hostnamectl set-hostname slave1
[root@85de83414bfe /]# bash
[root@slave1 /]#
```

3. 在slave2中:

```
[root@kmaster ~]# docker exec -it slave2 /bin/bash
[root@85de83414bfe /]# hostnamectl set-hostname slave2
[root@85de83414bfe /]# bash
[root@slave2 /]#
```

4.1.3 查看每个容器节点的ip并配置ip映射

1. 配置master容器的hosts文件

```
[root@master /]# vi /etc/hosts
```

2. 配置文件内容:

```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      master
172.17.0.4      slave1
172.17.0.3      slave2
```

3. 将hosts文件拷贝给slave1和slave2:

```
[root@master /]# scp /etc/hosts slave1:/etc/
[root@master /]# scp /etc/hosts slave2:/etc/
```

4.1.4 对容器内各节点进行免密登录授权

1. 在master节点生成秘钥:

```
[root@master /]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

```
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:rbwlxsWEc5Guic01OtiJh20ZStPy6ArVs9vp5aAmMLI root@master
The key's randomart image is:
+---[RSA 2048]----+
|          ..     |
|         ...     |
|       .o.o      |
|     .+ o*+      |
|    ..o/SO+.     |
|. o.  B=/o       |
| o.o ..oB.o      |
|E  .. o= O       |
|    .+o.= .      |
+----[SHA256]-----+
[root@master /]#
```

2. 将秘钥拷贝到master/slave1/slave2：

```
[root@master /]# ssh-copy-id master
[root@master /]# ssh-copy-id slave1
[root@master /]# ssh-copy-id slave2
```

注：请自行在slave1和slave2节点重复该操作，内容略...

4.1.5 将jdk和hadoop安装包从宿主机拷贝到master容器中：

```
[root@kmaster ~]# docker cp /opt/softwares/02_hadoop/hadoop-2.7.2.tar.gz master:/opt/
[root@kmaster ~]# docker cp /opt/softwares/01_jdk/jdk-8u144-linux-x64.tar.gz master:/opt/
[root@kmaster ~]#
```

去容器中查看文件是否存在

```
[root@kmaster ~]# docker exec -it master /bin/bash
[root@master /]# ls /opt/
hadoop-2.7.2.tar.gz
[root@master /]#
```

4.2 安装和配置jdk

1. 将jdk解压到/usr/local 目录下:

```
[root@master /]# tar -zxvf /opt/jdk-8u144-linux-x64.tar.gz -C /usr/local/src/
```

2. 重命名jdk文件夹为java

```
[root@master /]# cd /usr/local/src
[root@master src]# mv jdk1.8.0_144/ java
[root@master src]# ls
hadoop-2.7.2  java
[root@master src]#
```

3. 配置java环境变量:

```
[root@master src]# vim /root/.bash_profile
```

4. 配置内容：

```
export JAVA_HOME=/usr/local/src/java
export PATH=$PATH:$JAVA_HOME/bin
```

5. source 配置文件并查看java版本：

```
[root@master src]# vim /root/.bash_profile
[root@master src]# source /root/.bash_profile
[root@master src]# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
[root@master src]#
```

6. 将jdk拷贝到两个从节点:

```
[root@master src]# scp -r /usr/local/src/java/ slave1:/usr/local/src/
[root@master src]# scp -r /usr/local/src/java/ slave2:/usr/local/src/
```

7. 将/root/.bash_profile拷贝到两个从节点:

```
[root@master src]# scp /root/.bash_profile slave1:/root/
.bash_profile
                      100%  246   451.5KB/s   00:00
[root@master src]# scp /root/.bash_profile slave2:/root/
.bash_profile
```

8. 在两个从节点source配置文件并查看java版本:

```
[root@master src]# ssh slave1
Last login: Tue May 10 03:00:24 2022 from master
[root@4f11a6697318 ~]# source /root/.bash_profile
[root@4f11a6697318 ~]# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
[root@4f11a6697318 ~]# ssh slave2
Last login: Tue May 10 03:01:12 2022 from slave1
[root@36aea7d07044 ~]# source /root/.bash_profile
[root@36aea7d07044 ~]# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
[root@36aea7d07044 ~]#
```

4.3 安装和配置hadoop

1. 将hadoop解压到/usr/local 目录下:

```
[root@master /]# tar -zxvf /opt/hadoop-2.7.2.tar.gz -C /usr/local/src/
```

2. 重命名hadoop文件夹为hadoop

```
[root@master src]# ls
hadoop-2.7.2  java
[root@master src]# mv hadoop-2.7.2/ hadoop/
[root@master src]#
```

3. 配置hadoop环境变量，仅当前用户生效:

```
[root@master src]# vi /root/.bash_profile
```

配置内容：

```
export HADOOP_HOME=/usr/local/src/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

4. source 配置文件并查看hadoop版本：

```
[root@master src]# source /root/.bash_profile
[root@master src]# hadoop version
Hadoop 2.7.2
Subversion Unknown -r Unknown
Compiled by root on 2017-05-22T10:49Z
Compiled with protoc 2.5.0
From source with checksum d0fda26633fa762bff87ec759ebe689c
This command was run using /usr/local/src/hadoop/share/hadoop/common/hadoop-common-2.7.2.jar
[root@master src]#
```

5. 配置hadoop-env.sh：

```
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/hadoop-env.sh
export JAVA_HOME=/usr/local/src/java
```

6. 配置core-site.xml:

命令：

```
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/core-site.xml
```

配置内容：

```
<property>
  <!--namenode的URL地址(必须写)-->
  <name>fs.defaultFS</name>
  <value>hdfs://master:9000</value>
</property>
<property>
  <!--SequenceFiles中使用的读/写缓冲区的大小，单位为KB,131072KB默认为64M(该配置可选)-->
  <name>io.file.buffer.size</name>
  <value>131072</value>
</property>
<property>
  <!--hadoop临时文件路径(可选配置)-->
  <name>hadoop.tmp.dir</name>
  <value>/usr/local/src/hadoop/dfs/tmp</value>
</property>
```

7. 配置hdfs-site.xml:

命令

```
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/hdfs-site.xml
```

配置内容：

```
<property>
  <!--hadoop的副本数量，默认为3(必须写)-->
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <!--在本地文件系统所在的NameNode的存储空间和持续化处理日志(必须写)-->
  <name>dfs.namenode.name.dir</name>
  <value>/usr/local/src/hadoop/dfs/name</value>
</property>
<property>
  <!--在本地文件系统所在的DataNode的存储空间和持续化处理日志(必须写)-->
  <name>dfs.datanode.data.dir</name>
  <value>/usr/local/src/hadoop/dfs/data</value>
</property>
<property>
  <!--设置namenode线程，处理datanode发出rpc请求数量（可选配置）-->
  <name>dfs.namenode.handler.count</name>
  <value>100</value>
</property>
```

8. 配置mapred-site.xml:

命令：

```
[root@master ~]# cp /usr/local/src/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/src/hadoop/etc/hadoop/mapred-site.xml
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml
```

配置内容：

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

9. 配置yarn-site.xml:

命令：

```
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/yarn-site.xml
```

配置内容：

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
```

10. 配置slaves:

命令：

```
[root@master ~]# vi /usr/local/src/hadoop/etc/hadoop/slaves
```

配置内容：

```
master
slave1
slave2
```

11. 将配置文件拷贝被slave1和slave2：

```
[root@master ~]# scp -r /usr/local/src/hadoop slave1:/usr/local/src/
[root@master ~]# scp -r /usr/local/src/hadoop slave2:/usr/local/src/
[root@master ~]# scp /root/.bash_profile slave1:/root
[root@master ~]# scp /root/.bash_profile slave2:/root
```

12. 格式化namenode:

```
[root@master src]# hdfs namenode -format
22/05/10 04:38:10 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/172.17.0.2
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.2
...省略
22/05/10 04:38:10 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1249629373-172.17.0.2-1652157490900
22/05/10 04:38:10 INFO common.Storage: Storage directory /usr/local/src/hadoop/dfs/name has been successfully formatted.
22/05/10 04:38:11 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
22/05/10 04:38:11 INFO util.ExitUtil: Exiting with status 0
22/05/10 04:38:11 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/172.17.0.2
************************************************************/
[root@master src]#
```

13. 启动hadoop集群 查看守护进程

1. 启动集群：

```
[root@master src]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/src/hadoop/logs/hadoop-root-namenode-master.out
master: starting datanode, logging to /usr/local/src/hadoop/logs/hadoop-root-datanode-master.out
slave2: starting datanode, logging to /usr/local/src/hadoop/logs/hadoop-root-datanode-slave2.out
slave1: starting datanode, logging to /usr/local/src/hadoop/logs/hadoop-root-datanode-slave1.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:PYg0eTs4K0euuQtR0CEb+9OS2sdX9IgduGceaHV/ktU.
ECDSA key fingerprint is MD5:15:98:0f:46:60:15:ee:c7:73:7b:19:92:1a:24:cd:19.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/src/hadoop/logs/hadoop-root-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/src/hadoop/logs/yarn--resourcemanager-85de83414bfe.out
slave1: starting nodemanager, logging to /usr/local/src/hadoop/logs/yarn-root-nodemanager-slave1.out
slave2: starting nodemanager, logging to /usr/local/src/hadoop/logs/yarn-root-nodemanager-slave2.out
master: starting nodemanager, logging to /usr/local/src/hadoop/logs/yarn-root-nodemanager-master.out
```

2. 查看进程：

master：

```
[root@master src]# jps
1605 DataNode
2325 Jps
1766 SecondaryNameNode
1926 ResourceManager
2040 NodeManager
1389 NameNode
[root@master src]#
```

slave1：

```
[root@slave1 ~]# jps
666 DataNode
891 Jps
767 NodeManager
[root@slave1 ~]#
```

slave2:

```
[root@slave2 ~]# jps
570 DataNode
795 Jps
671 NodeManager
[root@slave2 ~]#
```

14. 创建文件测试hdfs和mapreduce：

1. 创建文件：

```
[root@master ~]# vi words.txt
```

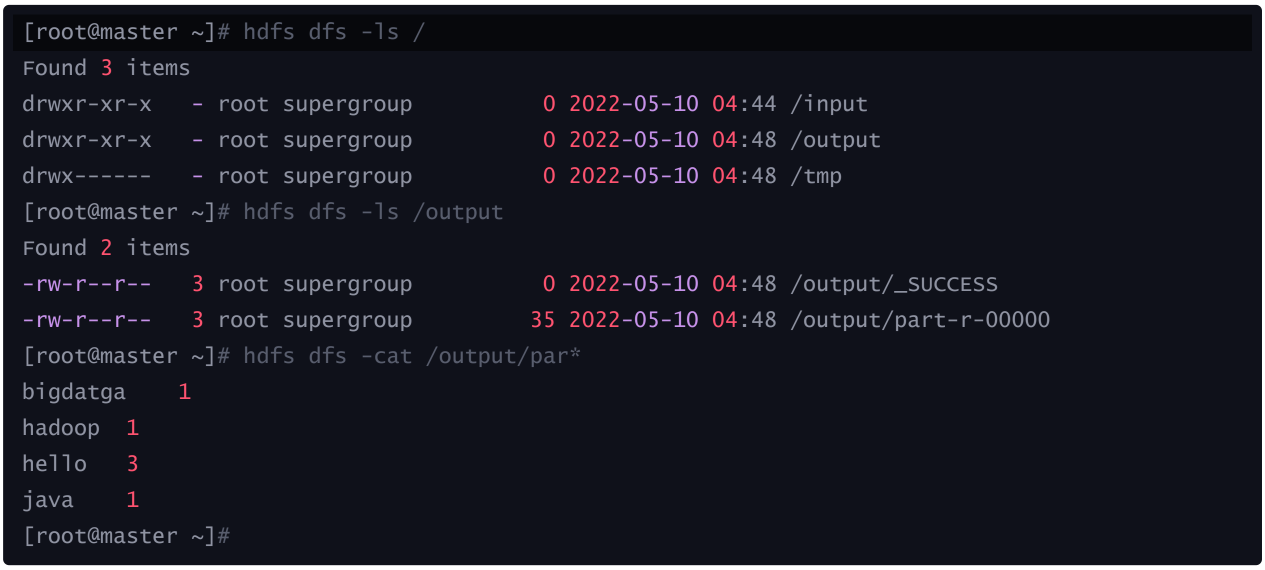
2. hdfs 创建文件夹并上传文件：

```
[root@master ~]# hdfs dfs -mkdir /input
[root@master ~]# hdfs dfs -put words.txt /input/
[root@master ~]# hdfs dfs -ls /input/
Found 1 items
-rw-r--r--   3 root supergroup         40 2022-05-10 04:44 /input/words.txt
[root@master ~]#
```

3. 运行hadoop自带的mapreduce测试yarn：

```
[root@master ~]# yarn jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount /input/words.txt /output/
```

4. 查看hdfs上的输出路径和文件内容：



```
[root@master ~]# hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - root supergroup          0 2022-05-10 04:44 /input
drwxr-xr-x   - root supergroup          0 2022-05-10 04:48 /output
drwx------   - root supergroup          0 2022-05-10 04:48 /tmp
[root@master ~]# hdfs dfs -ls /output
Found 2 items
-rw-r--r--   3 root supergroup          0 2022-05-10 04:48 /output/_SUCCESS
-rw-r--r--   3 root supergroup         35 2022-05-10 04:48 /output/part-r-00000
[root@master ~]# hdfs dfs -cat /output/par*
bigdatga    1
hadoop  1
hello   3
java    1
[root@master ~]#
```