

centos7.6搭建Hadoop2.8.5集群

1.基础环境

系统：centos7.6

硬件架构：aarch64

服务器：华为泰山2280v2

jdk：jdk-8u221-linux-arm64-vfp-hflt.tar.gz

集群规划

IP地址	主机名	角色
192.168.101.223	master	master, namenode
192.168.101.224	slave1	slave, datanode
192.168.101.225	slave2	slave, datanode

配置hosts文件，三个服务器均要做此操作。

```
[root@master ~]# vim /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.101.223  master
192.168.101.224  slave1
192.168.101.225  slave2
```

卸载系统自带的openJdk。

查看当前jdk信息

```
rpm -qa | grep java
```

然后用以下命令进行删除

```
rpm -e --nodeps jdkname
```

安装jdk，建立java目录

```
[root@master ~]# mkdir -p /usr/local/lib/java
[root@master ~]# tar -zxf jdk-8u221-linux-arm64-vfp-hflt.tar.gz -C
/usr/local/lib/java
```

配置环境变量

```
[root@master ~]# vim /etc/profile
export JAVA_HOME=/usr/local/lib/java/jdk1.8.0_221
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

查看当前jdk信息

```
[root@master java]# java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
```

2.关闭防火墙

查看防火墙状态

```
firewall-cmd --state
```

关闭防火墙

```
systemctl stop firewalld.service
```

禁止防火墙开机自启

```
systemctl disable firewalld.service
```

关闭selinux, 打开/etc/selinux/config文件,SELINUX=enforcing改为SELINUX=disabled

```
[root@master ~]# vim /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are
protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3.配置ssh免密登录

在每一台服务器上做如下操作。

```
[root@master ~]# ssh-keygen -t rsa #然后三个回车即可
[root@master ~]# ssh-copy-id master
[root@master ~]# ssh-copy-id slave1
[root@master ~]# ssh-copy-id slave2
```

三台服务器配置完成后，可以使用ssh检验免密登录是否成功。

```
[root@master ~]# ssh slave1
Last login: Fri Sep 27 14:28:40 2019 from 192.168.101.41
[root@slave1 ~]#
```

4.安装hadoop

下载hadoop文件

地址: <http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.8.5/hadoop-2.8.5.tar.gz>

hadoop安装目录为/opt/bigdata/hadoop-2.8.5

首先把文件解压到/opt/bigdata目录下

```
tar -zxvf hadoop-2.8.5.tar.gz -C /opt/bigdata
```

配置core-site.xml文件，内容如下

```
[root@master ~]# cd /opt/bigdata/hadoop-2.8.5/etc/hadoop/
[root@master hadoop]# vim core-site.xml
<configuration>
<!--配置hdfs文件系统的命名空间-->
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>

<!-- 配置操作hdfs的存冲大小 -->
  <property>
    <name>io.file.buffer.size</name>
    <value>4096</value>
  </property>

<!-- 配置临时数据存储目录 -->
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/opt/bigdata/hadoop-2.8.5/tmp</value>
  </property>
  <property>
    <name>hadoop.proxyuser.root.hosts</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.root.groups</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.oozie.hosts</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.oozie.groups</name>
    <value>*</value>
  </property>
  <property>
```

```

<name>hadoop.proxyuser.hue.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hue.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hive.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.hive.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.root.hosts</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.root.groups</name>
<value>*</value>
</property>
</configuration>

```

配置hdfs-site.xml, 内容如下

```

[root@master hadoop]# vim hdfs-site.xml
<configuration>
<!--配置副本数-->
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
<!--hdfs的元数据存储位置-->
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/opt/bigdata/hadoop-2.8.5/hdfs/name</value>
    </property>
<!--hdfs的数据存储位置-->
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/opt/bigdata/hadoop-2.8.5/hdfs/data</value>
    </property>
<!--hdfs的namenode的web ui 地址-->
    <property>
        <name>dfs.http.address</name>
        <value>master:50070</value>
    </property>
<!--hdfs的snn的web ui 地址-->
    <property>
        <name>dfs.secondary.http.address</name>
        <value>Master:50090</value>
    </property>
<!--是否开启web操作hdfs-->
    <property>
        <name>dfs.webhdfs.enabled</name>

```

```

        <value>true</value>
    </property>
<!--是否启用hdfs权限（acl）-->
    <property>
        <name>dfs.permissions</name>
        <value>>false</value> </property>
    <property>
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
    </property>
</configuration>

```

配置mapred-site.xml, 内容如下

```

[root@master hadoop]# cp mapred-site.xml.template mapred-site.xml
[root@master hadoop]# vim mapred-site.xml
<configuration>
<!--指定maoreduce运行框架-->
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value> </property>
<!--历史服务的通信地址-->
    <property>
        <name>mapreduce.jobhistory.address</name>
        <value>master:10020</value>
    </property>
<!--历史服务的web ui地址-->
    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>master:19888</value>
    </property>
</configuration>

```

配置yarn-site.xml, 内容如下

```

[root@master hadoop]# vim yarn-site.xml
<configuration>

<!-- Site specific YARN configuration properties -->
<!--指定resourcemanager所启动的服务器主机名-->
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
    </property>
<!--指定mapreduce的shuffle-->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
<!--指定resourcemanager的内部通讯地址-->
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8032</value>
    </property>
<!--指定scheduler的内部通讯地址-->
    <property>

```

```

        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
    </property>
<!--指定resource-tracker的内部通讯地址-->
    <property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>master:8031</value>
    </property>
<!--指定resourcemanager.admin的内部通讯地址-->
    <property>
        <name>yarn.resourcemanager.admin.address</name>
        <value>master:8033</value>
    </property>
<!--指定resourcemanager.webapp的ui监控地址-->
    <property>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>master:8088</value>
    </property>
</configuration>

```

配置hadoop-env.sh, 指定JAVA_HOME

```

[root@master hadoop]# vim hadoop-env.sh
export JAVA_HOME=/usr/local/lib/java/jdk1.8.0_221

```

配置yarn-env.sh, 指定JAVA_HOME

```

[root@master hadoop]# vim yarn-env.sh
export JAVA_HOME=/usr/local/lib/java/jdk1.8.0_221

```

配置mapred-env.sh, 指定JAVA_HOME

```

[root@master hadoop]# vim mapred-env.sh
export JAVA_HOME=/usr/local/lib/java/jdk1.8.0_221

```

至此, hadoop配置完毕, 将配好的hadoop分发到另外两台服务器。首先要在另外两台服务器建立/opt/bigdata目录

```

[root@master ~]# scp -r /opt/bigdata/hadoop-2.8.5 root@slave1:/opt/bigdata
[root@master ~]# scp -r /opt/bigdata/hadoop-2.8.5 root@slave2:/opt/bigdata

```

配置hadoop环境变量, 每个服务器都要操作

```

[root@master ~]# vim /etc/profile
export HADOOP_HOME=/opt/bigdata/hadoop-2.8.5
export PATH=$PATH:$HADOOP_HOME/bin

```

5.启动并验证hadoop集群

第一次启动集群, 需要格式化namenode, 操作如下:

```
[root@master ~]# hdfs namenode -format
```

格式化成功后则启动hadoop

```
[root@master hadoop]# start-dfs.sh
```

启动yarn

```
[root@master hadoop]# start-yarn.sh
```

web验证

打开浏览器，输入地址：<http://192.168.101.223:50070>