# Hadoop安装及搭建完全分布式

## 前言

腾讯云服务器，系统CentOS

三台服务器分别名称代号为：master slave1 slave2

## 环境配置

### jdk安装

在master机器节点(自行决定哪台机器作为master节点)进行操作：

1. 下载jdk文件，此处推荐华为镜像站的jdk(1.8.0)文件网址

wget <https://repo.huaweicloud.com/java/jdk/8u151-b12/jdk-8u151-linux-x64.tar.gz>

1. 新建java文件夹用于jdk解压，下载完之后,在usr文件夹下创建一个java文件夹

mkdir /usr/java

1. 解压jdk文件至刚刚创建的文件夹

tar -zxvf jdk-8u151-linux-x64.tar.gz -C /usr/java

1. 配置java环境变量

vim /etc/profile

1. 修改配置文件(随便找个位置加入以下内容)

export JAVA\_HOME=/usr/java/jdk1.8.0\_151

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

1. 使环境变量生效

source /etc/profile

1. 结果测试

java -version

如果能够返回java版本号则jdk安装成功

### 防火墙关闭

sudo systemctl stop firewalld.service

sudo systemctl disable firewalld.service

### 修改hosts文件

43.143.143.211 master

43.140.196.183 node1

43.143.132.125 node2

## 建立机器间互信

### 描述

由于hadoop必须机器之间免密登录，所以我们必须采取ssh密钥认证方式去保证机器登录。

机器a向机器b建立免密登录的流程：

* 机器a生成公钥
* 机器a将认证文件传给机器b
* 机器a成功单向建立与机器b的免密登录(即a可以免密登录b但b不能免密登录a,只需要在b上执行上述机器a的所有操作即可完成b到a的免密登录)

### **生成公钥**

ssh-keygen

遇到提示直接全回车就行，成功后会出现一个奇形怪状内容的矩形框，标头为RSA，成功之后会在 ~/.ssh文件夹下面看到id\_rsa id\_rsa.pub文件(为root用户登录情况下，可能会有authorized\_keys文件，问题不大)

跳转至~/.ssh文件夹下

cd ~/.ssh

### **传输认证文件**

传递至node1

传递时需要输入一次 slave1结点上的 root用户的登录密码

ssh-copy-id slave1

检查是否成功

ssh slave1

如果不需要输入密码即出现了日期信息，则传递至slave1成功

以同样的方式，传递至slave2

## 安装hadoop以及hadoop配置

### hadoop下载

此处推荐使用清华镜像的下载链接(纵享丝滑+1了)：

wget <https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz>

解压hadoop文件：

tar -zxvf hadoop-3.1.3.tar.gz -C /usr/hadoop

配置环境变量：

vim /etc/profile

将之前配置jdk时候的内容替换为:

export JAVA\_HOME=/usr/java/jdk1.8.0\_151

export HADOOP\_HOME=/usr/hadoop/hadoop-3.1.3

export PATH=$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$PATH

export PATH=$PATH:$HADOOP\_HOME/sbin

export HADOOP\_LOG\_DIR=/usr/hadoop/hadoop-3.1.3/logs

export YARN\_LOG\_DIR=$HADOOP\_LOG\_DIR

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

使环境变量生效：

source /etc/profile

测试：

hadoop version

### 配置hadoop

创建文件夹：

mkdir /usr/hadoop/hadoop-3.1.3/tmp

mkdir /usr/hadoop/hadoop-3.1.3/hdfs

mkdir /usr/hadoop/hadoop-3.1.3/hdfs/name

mkdir /usr/hadoop/hadoop-3.1.3/hdfs/data

cd /usr/hadoop/hadoop-3.1.3/etc/hadoop/

#### 修改core-site.xml(在文件末尾的 之间添加如下代码)：

<!--配置hdfs文件系统的命名空间-->

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoop.master:9000</value> <!--master处为主机名,9000为端口号-->

</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>file:/usr/hadoop/hadoop-3.3.4/tmp</value>

</property>

<property>

<name>hadoop.http.staticuser.user</name>

<!-- 指定静态角色，HDFS上传文件的时候需要 -->

<value>root</value>

</property>

<!--web console cors settings允许WebHDFS跨域操作-->

<property>

<name>hadoop.http.filter.initializers</name>

<value>org.apache.hadoop.security.HttpCrossOriginFilterInitializer</value>

</property>

<property>

<name>hadoop.http.cross-origin.enabled</name>

<value>true</value>

</property>

<property>

<name>hadoop.http.cross-origin.allowed-origins</name>

<value>\*</value>

</property>

<property>

<name>hadoop.http.cross-origin.allowed-methods</name>

<value>GET,POST,HEAD</value>

</property>

<property>

<name>hadoop.http.cross-origin.allowed-headers</name>

<value>X-Requested-With,Content-Type,Accept,Origin</value>

</property>

<property>

<name>hadoop.http.cross-origin.max-age</name>

<value>1800</value>

</property>

#### 修改hdfs-site.xml

<!--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>file:/usr/hadoop/hadoop-3.1.3/hdfs/name</value>

</property>

<!--hdfs的数据存储位置-->

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/hadoop/hadoop-3.1.3/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>0.0.0.0:50090</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>

#### 修改mapred-site.xml

<!--指定mapreduce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>

#### 修改yarn-site.xml

<!--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>

修改workers(hadoop3之前文件名为slaves)，直接将文件内容替换为

master

slave1

slave2

配置hadoop-env.sh、yarn-env.sh、mapred-env.sh，加入自己的jdk路径，即/usr/java/jdk1.8.0\_151(随便在文件找个空行加入以下内容)

export JAVA\_HOME=/usr/java/jdk1.8.0\_151

配置文件拷贝，发送指令： scp+ -r +待发送文件夹+ 用户名@主机名：另一台机器的文件接收路径

scp -r /usr/hadoop root@slave1:/usr

## 启动集群

### 启动hadoop

跳转目录

cd /usr/hadoop/hadoop-3.1.3/sbin

格式化namenode

hdfs namenode -format

启动所有服务：

start-all.sh

若出现很多ERROR的情况则进行下列操作：

cd /usr/hadoop/hadoop-3.1.3/sbin

修改start-dfs.sh和stop-dfs.sh文件，添加如下内容：

HDFS\_DATANODE\_USER=root

HADOOP\_SECURE\_DN\_USER=hdfs

HDFS\_NAMENODE\_USER=root

HDFS\_SECONDARYNAMENODE\_USER=root

修改start-yarn.sh和stop-yarn.sh文件，添加如下内容：

YARN\_RESOURCEMANAGER\_USER=root

HADOOP\_SECURE\_DN\_USER=yarn

YARN\_NODEMANAGER\_USER=root

重新执行代码：

start-all.sh

成功后输入jps指令查看情况

Jps

应该能看到六行信息包括如下内容:

NodeManager ResourceManager NameNode Jps DataNode SecondaryNameNode

### **查看页面**

http://master服务器的公网ip:50070/