# 第一步：搭建配置新的虚拟机

格式化之前先把tmp目录下所有与Hadoop有关的信息全部删除

rm -rf /tmp/hadoop-centos\*

开启之后jps只有Java的进程：sudo vi /etc/hosts 里面加 bogon

## 1.sudo赋权

Root用户 vi /etc/sudoers

/root n（查找下一个结果）

centos ALL=(ALL) NOPASSWD:ALL

## 修改ens33网卡（桥接模式静态IP）

创建软链接

ln –s /etc/sysconfig/network-scripts/ifcfg-ens33 ens33

修改配置文件

vi /etc/sysconfig/network-scripts/ifcfg-ens33

|  |
| --- |
| TYPE=Ethernet  BOOTPROTO=static  DEFROUTE=yes  PEERDNS=yes  PEERROUTES=yes  IPV4\_FAILURE\_FATAL=no  IPV6INIT=yes  IPV6\_AUTOCONF=yes  IPV6\_DEFROUTE=yes  IPV6\_PEERDNS=yes  IPV6\_PEERROUTES=yes  IPV6\_FAILURE\_FATAL=no  IPV6\_ADDR\_GEN\_MODE=stable-privacy  NAME=ens33  UUID=51248a5b-aece-4777-ab51-7b5e61602180  DEVICE=ens33  ONBOOT=yes  IPADDR=192.168.12.206  NETMASK=255.255.255.0  GATEWAY=192.168.12.1  DNA1=219.141.136.10  DNS2=202.106.196.115  DNS3=8.8.8.8  DNS4=114.114.114.114 |

改完后重启network服务

sudo systemctl restart network

虚拟机上右键-->设置-->网络适配器-->桥接模式

检测是否连接网络 ping www.baidu.com

## 修改主机名

sudo vi /etc/hostname

原来内容全部删除 写入s126

重启虚拟机

## 安装JDK和Hadoop

1. 上传文件包
2. 解压 tar –zxvf \*\*\*\*
3. 创建软链接 ln –s \*\*\* hadoop

tar -zxvf jdk-8u121-linux-x64.tar.gz

ln -s jdk1.8.0\_121 java

rm -rf jdk-8u121-linux-x64.tar.gz

tar -zxvf hadoop-2.8.0.tar.gz

ln -s hadoop-2.8.0 hadoop

rm -rf hadoop-2.8.0.tar.gz

4.配置环境变量

sudo vi /etc/profile

#java

JAVA\_HOME=/home/centos/java

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

#hadoop

HADOOP\_HOME=/home/centos/hadoop

Export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

vi ~/hadoop/etc/hadoop/hadoop-env.sh

#java

export JAVA\_HOME=/home/centos/java

1. 刷新配置文件

source /etc/profile

## 关闭防火墙

sudo systemctl stop firewalld关闭防火墙

sudo systemctl start firewalld开启防火墙

sudo systemctl disable firewalld禁用防火墙（开机不启动）

sudo systemctl enable firewalld使用防火墙（开机启动）

sudo systemctl status firewalld查看防火墙状态

# 第二步：Hadoop集群搭建

## 独立模式

1. 将hadoop的jar包上传服务器
2. 解压tar包

tar –zxvf \*\*\*\*

1. 创建软连接

ln –s \*\*\* hadoop

1. 配置环境变量

[vi /etc/profile]

最下面加上

#hadoop

HADOOP\_HOME=/home/centos/hadoop

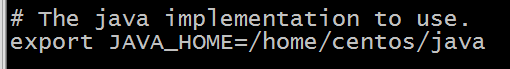
export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

最后保存文件加载环境变量配置文件

source /etc/profile

1. 修改hadoop环境变量配置文件配置jdk路径

[$HADOOP\_HOME/etc/hadoop/hadoop-env.sh]



1. 测试

hadoop version

默认hadoop就是独立模式。

特点：运行任务是一个java进程，就相当于运行一个java程序

## 2.伪分布式模式配置

**修改hdfs配置文件**

vi ~/hadoop/etc/hadoop/core-site.xml

cd ~/hadoop/etc/hadoop

[core-site.xml]

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.12.126:9000</value>

</property>

</configuration>

vi ~/hadoop/etc/hadoop/hdfs-site.xml

[hdfs-site.xml]

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

**设置ssh免密登录**

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 600 ~/.ssh/authorized\_keys

检测方式：ssh localhost/主机真实ip

**格式化文件系统**

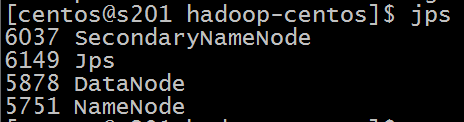
hadoop namenode -format

**启动文件系统**

jps 检测Java开启的进程

start-dfs.sh 启动文件系统（启动了三个java进程）

jps 再次检测Java开启的进程（应该是4个）



**配置yarn资源管理器**

cd ~/hadoop/etc/hadoop

cp mapred-site.xml.template mapred-site.xml

vi mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

vi yarn-site.xml

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

start-dfs.sh && start-yarn.sh && jps

## 完全分布式

配置/etc/hosts文件（本地静态DNS）

首行加上 : bogon

增加:

192.168.12.201 s201

192.168.12.202 s202

192.168.12.203 s203

1. **设置ssh免密登录：**

**(现在别的主机ssh 主机IP 然后把宿主机复制过去的权限600)**

效果：从主机上能通过ssh免密登录

目的：在主机上通过ssh执行命令

设置的方式：

首先给主机设置ssh免密登录，将主机的公钥(id\_rsa.pub)通过scp命令发送到其他 主机，在其他主机上将获取到的公钥追加到~/.ssh/authorized\_keys。

主机：

ssh-keygen –t rsa –P '' –f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 600 ~/.ssh/authorized\_keys

scp ~/.ssh/id\_rsa.pub [centos@192.168.12.202:~/.ssh/id\_rsa.pub.s201](mailto:centos@192.168.12.202:~/.ssh/id_rsa.pub.s201)

scp ~/.ssh/id\_rsa.pub [centos@192.168.12.203:~/.ssh/id\_rsa.pub.s201](mailto:centos@192.168.12.203:~/.ssh/id_rsa.pub.s201)

s202：

s203：

cat ~/.ssh/id\_rsa.pub.s201 >> ~/.ssh/authorized\_keys

**2.shell脚本的操作：**

主机写分发脚本cpTo，查看脚本xcall。赋予执行权限。移动到/usr/local/bin目录下，以便我们能在任何地方使用脚本

**[cpTo]**

#脚本的目的是：修改完了主机的文件以后，可以通过它分发到其他主机，实现同步

#脚本的用法：cpTo 文件（绝对路径/相对路径）

#!/bin/bash

#获取参数的绝对路径

dirname=`cd $(dirname $1); pwd`

basename=`basename $1`

echo ${dirname}/${basename}

for i in s202 s203

do

echo "=================$i ${dirname}/${basename}=================="

#如果其他机器没有父目录创建父目录

ssh $i mkdir -p ${dirname}

#远程拷贝文件

scp $1 ${i}:${dirname}/${basename}

done

**[xcall.sh]**

#作用是通过ssh远程运行命令，比如查看jps，在比如删除/tmp

#用法：xcall 命令

#!/bin/bash

for host in s201 s202 s203

do

echo "====================echo $host $@==========================="

#ssh远程执行命令

#$@获取所有参数

#source /etc/profile 因为ssh远程执行命令不去加载/etc/profile，所以很多命令没法用

ssh $host "source /etc/profile && $@"

Done

**3.软件安装的操作**

安装jdk、hadoop、配置环境变量JAVA\_HOME、HADOOP\_HOME，可以通过xcall脚本来 同步安装

在主机上修改hosts文件，给主机配置所有机器的dns服务，使用cpTo脚本分发到其他 机器，让其他机器也有dns服务

**注意**：/etc/hosts是root用户的文件，所以需要切换到root执行

dns服务： 可以将域名解析为ip地址。比如ping s201，解析成ping 192.168.12.201

**4.配置文件操作**

在主机上修改hadoop的配置文件：

[Hadoop-en.sh] [core-site.xml] [hdfs-site.xml] [mapred-site.xml] [yarn-site.xml] [slaves]

通过cpTo脚本进行分发（可以分发上一级目录来全部分发）

1、[hadoop-env.sh]配置javahome

2、[core-site.xml]配置连接url

[core-site.xml]

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://s201:9000</value>

</property>

</configuration>

3、[hdfs-site.xml]配置的是副本数

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

</configuration>

4、[mapred-site.xml]配置运行map任务采用的集群资源管理框架

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

5、[yarn-site.xml]配置RM的节点和数据传输方式

<configuration>

<property>

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

<value>s126</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

6、[slaves]配置datanode节点和NM节点

s201

s202

s203

1. **主机格式化文件系统**

只需要在主机格式化文件系统（xcall删除数据目录/tmp/hadoop\*，日志文件/logs/\*）

启动hdfs文件系统和yarn框架

xcall查看启动是否成功/看web

# 第三步：配置项目

1. 将要统计的文件（wordcount.txt）上传到hdfs

hdfs dfs –put 本地文件 hdfs目录

hdfs dfs -put wordCount2.txt /user

1. 运行任务

hadoop jar jar包 com.zdjy.bigdata.wordCount.hadoop.WordCountApp hdfs输 入文件

hdfs输出目录

hadoop jar wordCount.jar com.zdjy.WordCountApp /user/wordCount2.txt /saa

1. 查看结果

hdfs dfs –cat hdfs输出目录下的文本文件 (hdfs dfs -cat /saa/\*)

1. 关闭文件系统

hdfs dfs -mkdir /user

hdfs dfs -ls -R /

hdfs -dfs -put wordCount2.txt /user

hdfs dfs -put wordCount2.txt /user

hdfs dfs -ls -R /

hadoop jar wordCount-0.0.1-SNAPSHOT.jar

com.zdjy.bigdata.wordCount.hadoop.WordCountApp

/user/wordCount2.txt /out

hdfs dfs -cat /out/\*

5.在浏览器上看web（sudo systemctl stop firewalld关闭防火墙）

<http://192.168.12.206:50070>

文件系统：50070

yarn：8088

# ssh免密登录与手动单独启动每一个进程

设置ssh免密登录的原因：我们start-dfs.sh/start-yarn.sh他们都是通过ssh来远程控制每一台主机的进程的启动

[start-dfs.sh]

if [ -n "$HADOOP\_SECURE\_DN\_USER" ]; then

echo \

"Attempting to start secure cluster, skipping datanodes. " \

"Run start-secure-dns.sh as root to complete startup."

else

"$HADOOP\_PREFIX/sbin/hadoop-daemons.sh" \

--config "$HADOOP\_CONF\_DIR" \

--script "$bin/hdfs" start datanode $dataStartOpt

fi

[Hadoop-daemons.sh]

exec "$bin/slaves.sh" --config $HADOOP\_CONF\_DIR cd "$HADOOP\_PREFIX" \; "$bin/hadoop-daemon.sh" --config $HADOOP\_CONF\_DIR "$@"

[slaves.sh]

for slave in $SLAVE\_NAMES ; do

ssh $HADOOP\_SSH\_OPTS $slave $"${@// /\\ }" \

2>&1 | sed "s/^/$slave: /" &

if [ "$HADOOP\_SLAVE\_SLEEP" != "" ]; then

sleep $HADOOP\_SLAVE\_SLEEP

fi

done

我们手动启动每一个进程

主机启动namenode

hadoop-daemon.sh start namenode

所有机器手动启动datnode

hadoop-daemon.sh start datanode

在主机上手动启动datanode

ssh s202 “source /etc/profile && hadoop-daemon.sh start datanode”

在主机上脚本启动所有datanode

hadoop-daemons.sh start datanode

在主机启动secondarynamenode

主机启动resourcemanager

yarn-daemon.sh start resourcemanager

所有机器启动nodemanager

yarn-daemon.sh start nodemanager

在主机启动所有的nodemanager

yarn-daemons.sh start nodemanager

start-dfs.sh=NN,DNs,2NN

start-yarn.sh=RM,NMs

start-all.sh= start-dfs.sh && start-yarn.sh