**Hadoop安装与配置**

*提示：*

**特别注意，如果安装的hadoop版本是3.xx，那么一定要用非root用户安装，否则需要改动hadoop脚本，不利于维护。**

# 环境配置：

测试安装节点：

192.168.159.132 master

192.168.159.134 node1

192.168.159.137 node2

jdk-8u101-Linux-x64.gz (Java )

hadoop-2.7.3.tar.gz (Hadoop 包)

# 安装步骤：

1. 修改各个虚拟机的hostname和host
2. 创建用户组和用户 hadoop:hadoop
3. 配置虚拟机网络，使虚拟机系统之间以及和host主机之间可以通过相互ping通。
4. 安装jdk和配置环境变量，检查是否配置成功
5. 配置**hadoop用户**的ssh，实现节点间的无密码登录 ssh node1/2指令验证时候成功
6. master配置hadoop，并将hadoop文件传输到node节点
7. 配置环境变量，并启动hadoop，检查是否安装成功，执行wordcount检查是否成功。

## 修改虚拟机的hostname和hosts文件

以master上机器为例，打开终端，执行如下的操作，把hostname修改成master，hosts修改成如下所示的样子：

#修改hostname的指令：

sudo vi /etc/hostname

#修改hosts指令：

sudo vi /etc/hosts

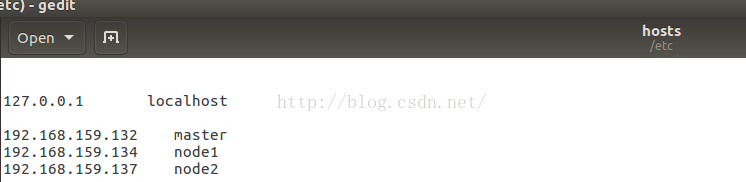
#将以下内容添加到hosts中

192.168.159.132 master

192.168.159.134 node1

192.168.159.137 node2

如下图所示:



同样地，在node1和node2机器上做相似的操作，分别更改主机名为node1和node2，然后把hosts文件更改和master一样。

## 创建用户和用户组（所有节点）

1. 创建hadoop用户组

sudo sudo addgroup hadoop

2. 创建hadoop用户

sudo adduser -ingroup hadoop hadoop

3. 给hadoop用户添加权限，打开/etc/sudoers文件

sudo vi /etc/sudoers

按回车键后就会打开/etc/sudoers文件了，给hadoop用户赋予root用户同样的权限

在

root  ALL=(ALL:ALL)  ALL

下添加

hadoop  ALL=(ALL:ALL)  ALL

## 检验各个主机之间能否连通

各节点切换到新建的hadoop用户，后面都以hadoop用户操作。

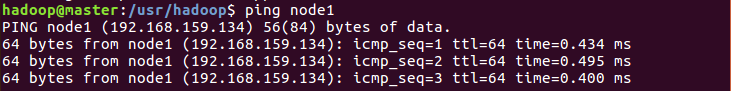
su hadoop

验证节点能否ping通

ping +主机名

分别在各个主机上执行上述指令，看是否能与其他主机连通。

出现下图代表能够连通：



如果都成功ping通，进行下面的操作。

## 安装jdk和配置环境变量

分别在每台主机上安装jdk，并配置环境变量。（嫌麻烦的前面可以安装完jdk后再克隆）

1）下载jdk安装包（自行百度），并将安装包拖入到虚拟机当中

2）通过cd命令进入到安装包的当前目录，利用如下命令进行解压缩。

tar -zxvf jdk.....(安装包名称)

3）利用如下命令将解压后的文件夹移到/usr目录下

#注意，这样移动到/usr以后就没有jdk1.8...这个目录了，是将这个目录下的所有文件全部移动到/usr/java下，

mv jdk1.8...(文件夹名称) /usr/java

4）配置环境变量

sudo vi /etc/profile

在末尾加上四行：

#java

export JAVA\_HOME=/usr/java

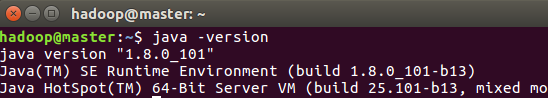
export JRE\_HOME=/usr/java/jre

export CLASSPATH=$JAVA\_HOME/lib

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

输入如下命令使配置生效：source /etc/profile

查看配置是否成功，



出现如上信息说明java配置成功。

## 配置SSH

（实现节点间的无密码登录）

本人在这一步经常出错，莫名其妙的错误，网上也找不到资料。需要自己多实验几次。

下面的 1.2.3在所有主机上都要做

1..安装ssh

sudo apt-get install openssh-server

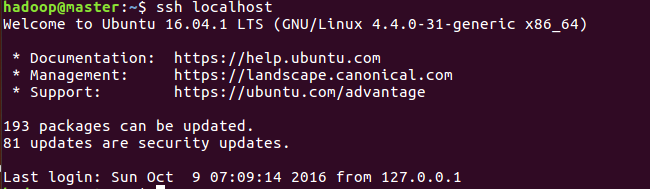
已有ssh或者安装成功了的输入命令

ps -e | grep ssh

验证SSH是否成功安装输入

ssh localhost

出现以下提示说明安装成功



3.生成密钥Pair

ssh-keygen -t rsa

输入之后一直选择enter即可。生成的秘钥位于 ~/.ssh文件夹下。可用cd 命令进入查看。

4.在master上，导入authorized\_keys

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

5.远程无密码登录(把master上的authorized\_keys拷贝到其他主机的相应目录下)

#进入master的.ssh目录,执行复制操作

scp authorized\_keys hadoop@node1:~/.ssh/

scp authorized\_keys hadoop@node2:~/.ssh/

修改各台主机上authorized\_keys文件的权限：

所有机器上，均执行命令：

chmod 600 .ssh/authorized\_keys

完成之后，在master上执行下面操作，检查免密码登录是否成功。

ssh node1(node2)

## master配置hadoop,

（配置完后将master的hadoop文件传送给各节点）

1）解包移动

#解压hadoop包

tar -zxvf hadoop...

#将安装包移到/usr目录下

mv hadoop... /usr/hadoop

2）新建文件夹

#在/usr/hadoop目录下新建如下目录(root)

sudo cd /usr/hadoop

sudo mkdir /dfs

sudo mkdir /dfs/name

sudo mkdir /dfs/data

sudo mkdir /tmp

3) 配置文件：hadoop-env.sh(文件都在/usr/hadoop/etc/hadoop中)

修改JAVA\_HOME值（export JAVA\_HOME=/usr/java）

4) 配置文件：yarn-env.sh

修改JAVA\_HOME值（export JAVA\_HOME=/usr/java）

5) 配置文件：slaves

将内容修改为：

node1

node2

6) 配置文件：core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://master:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

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

<description>Abase for other temporary directories.</description>

</property>

</configuration>

7) 配置文件：hdfs-site.xml

<configuration>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>master:9001</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/hadoop/dfs/name</value>

</property>

<property>

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

<value>file:/usr/hadoop/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

8) 配置文件：mapred-site.xml

先创建然后编辑

cp etc/hadoop/mapred-site.xml.template etc/hadoop/mapred-site.xml

vi etc/hadoop/mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>master:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>master:19888</value>

</property>

</configuration>

9) 配置文件：yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-servi ces</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-servi ces.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>master:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>master:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>master:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>master:8088</value>

</property>

<!-- 如果不配置，会报错，在yarn任务里可以看到如下的报错， 错误: 找不到或无法加载主类 org.apache.hadoop.mapreduce.v2.app.MRAppMaster。解决方法是执行hadoop classpath命令，然后复制结果到这里 -->

<property>

<name>yarn.application.classpath</name>

<value>${hadoop classpath}</value>

</property>

<!-- 如果不配置，会报错 The auxService:mapreduce\_shuffle does not exist -->

<property>

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

      <value>mapreduce\_shuffle</value>

</property>

</configuration>

10) 将hadoop传输到node1和node2 usr/hadoop目录,（如果传输时报错说 ：权限拒绝，先把文件传送到非/usr目录下，然后在node上把这个文件再移动到/usr/hadoop）

scp -r /usr/hadoop hadoop@node1:/usr/hadoop

## 配置环境变量

#编辑/etc/profile

sudo vi /etc/profile

#以上已经添加过java的环境变量,在其后边添加就可以

#hadoop

export HADOOP\_HOME=/opt/Hadoop/hadoop-2.7.2

export PATH=$PATH:$HADOOP\_HOME/sbin

export PATH=$PATH:$HADOOP\_HOME/bin

使文件生效

source /etc/profile

## pid路径和日志路径

pid路径配置

默认放在/tmp下，带来的问题是万一被删除，就没法通过stop命令来停止集群

编辑hadoop-env.sh 配置属性HADOOP\_PID\_DIR

日志路径配置

编辑hadoop-env.sh 配置属性HADOOP\_LOG\_DIR

## 启动hadoop，检查是否安装成功

格式化namenode

hdfs namenode -format

启动hadoop,进入hadoop安装目录

start-all.sh

启动后分别在master, node下输入jps查看进程

看到下面的结果，则表示成功。

Master:



Slave:



## 测试是否可用

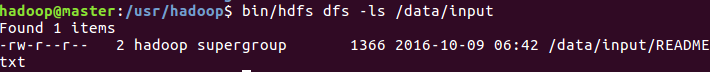
进入本地hadoop目录（/usr/hadoop）

1、  bin/hdfs dfs -mkdir -p /data/input在虚拟分布式文件系统上创建一个测试目录/data/input

2、  hdfs dfs -put README.txt  /data/input  将当前目录下的README.txt 文件复制到虚拟分布式文件系统中

3、  bin/hdfs dfs-ls /data/input    查看文件系统中是否存在我们所复制的文件

如图操作：



3、  运行如下命令向hadoop提交单词统计任务

进入jar文件目录，执行下面的指令。

hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /data/input /data/output/result

查看result，结果在result下面的part-r-00000中

hdfs dfs -cat /data/output/result/part-r-00000

# Hadoop权限管理

参考：<http://hadoop.apache.org/docs/r1.0.4/cn/hdfs_permissions_guide.html>

非常重要的一点是：hdfs的客户端用户直接从宿主系统中获取。

# 异常

## 主节点没有namenode的处理

解决办法是 初始化namenode

bin/hadoop namenode -format

stop-all.sh

start-all.sh

## 库加载失败

执行hdfs命令报错：

Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

执行命令查看库也没有：

hadoop checknative -a

原因：没有编译导致缺少链接库

解决方式：

方式一：针对本项目，到沈阳的机器上获取库，然后放到hadoop的lib的native下

方式二：使用第三方库（http://dl.bintray.com/sequenceiq/sequenceiq-bin/）

方式三：（未验证）在文件hadoop-env.sh中增加：

export HADOOP\_OPTS="-Djava.library.path=${HADOOP\_HOME}/lib/native"

方式四：可能是gcc版本与所需版本不一致。