1. 环境

Windows 10 企业版

VMware workstation 12

CentOS-7-x86\_64-Everything-1708

Hadoop2.9.0

1. 步骤

1.**安装虚拟机**

1.1官方镜像站下载hadoop2.9.0：<http://mirror.bit.edu.cn/apache/hadoop/common/>

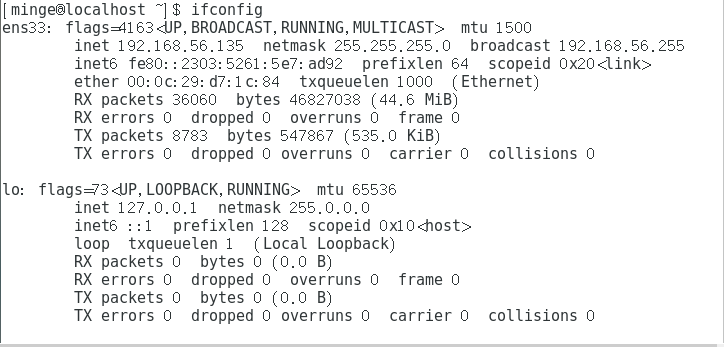
1.2下载并安装VMware WorkStation 12

1.3在VMware上安装三个CentOS虚拟机，分别命名为hserver1、hserver2、hserver3.安装过程较为简单，有几个点提一下：

（1）

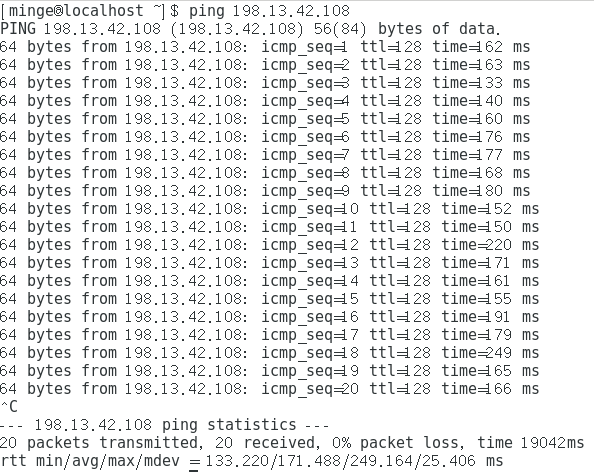
（2）安装时候一定要记得打开网络连接（忘记截图了）

1.4 安装完成，ifconfig测一下



看到hserver1的ip（内部ip，更多知识请百度NAT）是192.168.56.135

1.5 测试能否与主机ping通：



三台虚拟机的内部ip分别为：

hserver1 192.168.56.135

hserver2 192.168.56.137

hserver3 192.168.56.139

1.6 用hostname命令把三台虚拟机主机名分别改成我们想要的：

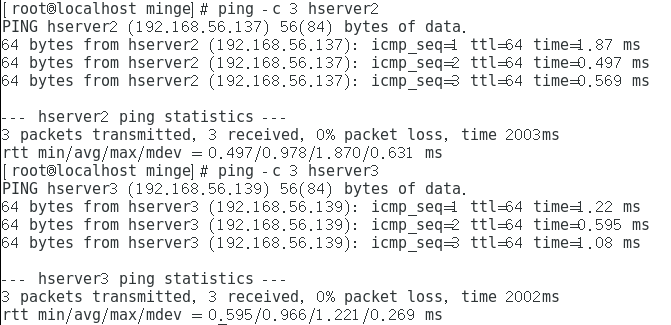
hostname hserver1

hostname hserver2

hostname hserver3

(如果当前登录的用户在sudoer文件中，需要在hostname命令前加sudo。否则需要切换到root用户来执行此命令：su root)

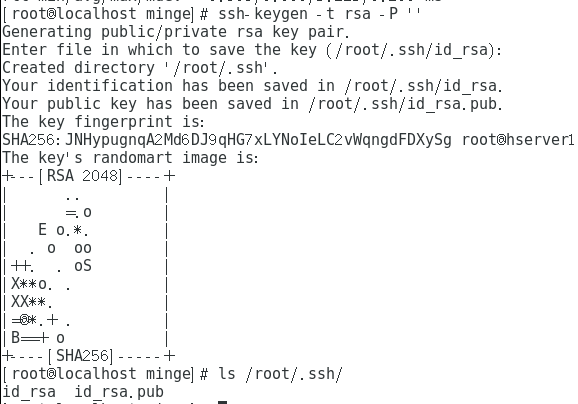
改完后检测一下三台虚拟机是否互相ping得通。以hserver1为例：



2.**配置ssh免密登录**

2.1给虚拟机生成密钥文件：ssh-keygen –t rsa –P ‘’

然后查看当前用户名文件夹下是否生成了一个ssh文件（应该一共有两个）：

ls /root/.ssh/（我是在root用户名下操作的）

2.2.将三台虚拟机上的公钥文件（id\_rsa\_pub）中的内容都放到hserver1的authorized\_key中，然后将这个文件复制到hserver2和hserver3中（公钥密码体系，给对方发消息之前要先用对方的公钥加密）

三个虚拟机中id\_rsa\_pub文件中内容：

hserver1:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD19wJGnAp4pRohLWPeBJxF8Y5fCgY+foVTGb8nn74ULAobCzh8y5fKalWcG8wGmGpqzn906Gl8P98VEoD2Z4iFQLD2YWlRaJVTJxEfsTfyAjYEVplOioV9r7I6gRs9dEbsJXuPcwBnDVIVHVjimZE9K3f5BWzGMPh2ZmTZdUAgz1zzZnm/4eBqyQwciZ2z6y2Uycch47E7ZeemS/dCEzQoXnBabHMS7oABiKheYHumQ70KKuRuSBqjHQp5ABufvrRlCwU6BHs8Z0js9mOHu6CdFMv32NbsmUaVMg+WAeIhcQ4UyaL2yLJvIkMHu3kc9RyB+xxuzT4XLj/jxy/8TptP root@hserver1

hserver2:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDDEzJME5Jy7a02eIu2agGYpXTbZt1NNKVYFOCh3mU4plednmfXR0c3ZWSD36Au4cnb2SvsKnGDaIdCrAfQOsP5uq1NE9KhzrC69HTCkRe64KRg8lRRFZn4km6my23BFD/NKKpSSrZS3EI8YKGWF2COWTyUUyhCJKTj6R4sjD8nKl9s7XlS0DZjip8wgPnt7h7cugZOGOrKdZ936dLSD85kAvejSz0AcHNlFjRrx/wCCENIsLxKXS0XaBQ9bjZnQD/OWywcAJ9Eb1cmhyK8TYctav4I/9RefQSib4UQ0z+eXrZYrMKCyofdN05GSNiAh3M9gVsDbPBzT4e2Quf21o+Z root@hserver2

hserver3:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC1ygg4lm5CSxRHaphu9YilKvUT/O/6+jnueqpz604d0zUit5JIbfw2eRq5vooSBkTAZM7Y0AtroKQl1uBaJ20krJrAK660nGyvRx94e5Fmj+OyvFnRdC7f91+pFKkjBzfmdoBekfjgF1ABW39WwvwGuihDlvGXyVh86tMHNtPDY/TeqBm5g3ubuFbjY+IpgfTgW1PTVG7RF/BWsGZ6g5yNXQdiQwf7RtOWEDHXtYJEgTXvhAxp//bI0ay+61BluM5VRyTuYgXN+T2R4jvxbxaobPPitLk7AoDJCPHVC2H0BzZAgTQM37oczLSkjYhZCtsYuVq39f0RYJRRGsoN7DTn root@hserver3

authorized\_key中内容：



然后将这个文件分别复制到hserver2、hserver3同一个文件夹中去（可以用scp命令，或者winscp软件）

3.安装java环境（对三个节点均执行下面操作）

3.1下载jdk，放到/opt下

3.2 tar –zxvf 解压jdk到/usr/local文件夹下

3.3 vim /etc/profile，在最后面加入：

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

export CLASSPATH=$:CLASSPATH:$JAVA\_HOME/lib/

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

其中第一行JAVA\_HOME的路径为jdk存放路径。

3.4 source /etc/profile

3.5java –version和javac –version来检验是否安装成功。

4.安装hadoop集群

对hserver1（主节点）进行如下操作：

4.1 将下载好的hadoop-2.9.0.tar.gz放到/opt下

4.2 tar –zxvf 到/opt

4.3 vim /etc/profile ，在最后面添加如下内容：

export HADOOP\_HOME=/opt/hadoop-2.9.0/

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

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

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_ROOT\_LOGGER=INFO,console

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

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

其中HADOOP\_HOME为hadoop安装位置

4.4新建几个目录

在/root目录下新建几个目录，复制粘贴执行下面的命令：

mkdir /root/hadoop

mkdir /root/hadoop/tmp

mkdir /root/hadoop/var

mkdir /root/hadoop/dfs

mkdir /root/hadoop/dfs/name

mkdir /root/hadoop/dfs/data

4.5 修改etc/hadoop中的一系列配置文件

修改/opt /hadoop-2.9.0/etc/hadoop目录内的一系列文件。

4.5.1 修改core-site.xml

修改/opt /hadoop-2.9.0/etc/hadoop /core-site.xml文件

在<configuration>节点内加入配置:

<property>

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

<value>/root/hadoop/tmp</value>

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

</property>

<property>

<name>fs.default.name</name>

<value>hdfs://hserver1:9000</value>

</property>

</configuration>

4.5.2 修改hadoop-env.sh

修改/opt /hadoop-2.9.0/etc/hadoop /hadoop-env.sh文件

将export JAVA\_HOME=${JAVA\_HOME}

修改为：

export JAVA\_HOME=/opt/java/jdk1.8.0\_121

说明：修改为自己的JDK路径

4.5.3 修改hdfs-site.xml

修改/opt /hadoop-2.9.0/etc/hadoop /hdfs-site.xml文件

在<configuration>节点内加入配置:

<property>

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

<value>/root/hadoop/dfs/name</value>

<description>Path on the local filesystem where theNameNode stores the namespace and transactions logs persistently.</description>

</property>

<property>

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

<value>/root/hadoop/dfs/data</value>

<description>Comma separated list of paths on the localfilesystem of a DataNode where it should store its blocks.</description>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

<description>need not permissions</description>

</property>

说明：dfs.permissions配置为false后，可以允许不要检查权限就生成dfs上的文件，方便倒是方便了，但是你需要防止误删除，请将它设置为true，或者直接将该property节点删除，因为默认就是true。

4.5.4 新建并且修改mapred-site.xml

在该版本中，有一个名为mapred-site.xml.template的文件，复制该文件，然后改名为mapred-site.xml，命令是：

cp /opt/hadoop/hadoop-2.8.0/etc/hadoop/mapred-site.xml.template /opt/hadoop/hadoop-2.8.0/etc/hadoop/mapred-site.xml

修改这个新建的mapred-site.xml文件，在<configuration>节点内加入配置:

<property>

<name>mapred.job.tracker</name>

<value>hserver1:49001</value>

</property>

<property>

<name>mapred.local.dir</name>

<value>/root/hadoop/var</value>

</property>

<property>

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

<value>yarn</value>

</property>

4.5.5 修改slaves文件

修改/opt /hadoop-2.9.0/etc/hadoop /slaves文件，将里面的localhost删除，添加如下内容：

hserver2

hserver3

4.5.6 修改yarn-site.xml文件

修改/opt /hadoop-2.9.0/etc/hadoop /yarn-site.xml文件，

在<configuration>节点内加入配置(注意了，内存根据机器配置越大越好，我这里只配2个G是因为机器不行):

<property>

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

<value>hserver1</value>

</property>

<property>

<description>The address of the applications manager interface in the RM.</description>

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

<value>${yarn.resourcemanager.hostname}:8032</value>

</property>

<property>

<description>The address of the scheduler interface.</description>

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

<value>${yarn.resourcemanager.hostname}:8030</value>

</property>

<property>

<description>The http address of the RM web application.</description>

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

<value>${yarn.resourcemanager.hostname}:8088</value>

</property>

<property>

<description>The https adddress of the RM web application.</description>

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

<value>${yarn.resourcemanager.hostname}:8090</value>

</property>

<property>

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

<value>${yarn.resourcemanager.hostname}:8031</value>

</property>

<property>

<description>The address of the RM admin interface.</description>

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

<value>${yarn.resourcemanager.hostname}:8033</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.scheduler.maximum-allocation-mb</name>

<value>2048</value>

<discription>每个节点可用内存,单位MB,默认8182MB</discription>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>2.1</value>

</property>

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>2048</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

</property>

说明：yarn.nodemanager.vmem-check-enabled这个的意思是忽略虚拟内存的检查，如果你是安装在虚拟机上，这个配置很有用，配上去之后后续操作不容易出问题。如果是实体机上，并且内存够多，可以将这个配置去掉。

4.6格式化namenode

cd   /opt /hadoop-2.9.0/bin

 ./hadoop  namenode  -format

4.7 启动dfs和yarn：

cd /opt/Hadoop-2.9.0/sbin

./start-dfs.sh

./start-yarn.sh

4.8 查看集群是否启动成功

jps

hserver1显示：

　　SecondaryNameNode

　　ResourceManager

　　NameNode

hserver2和hserver3显示

　　NodeManager

　　DataNode