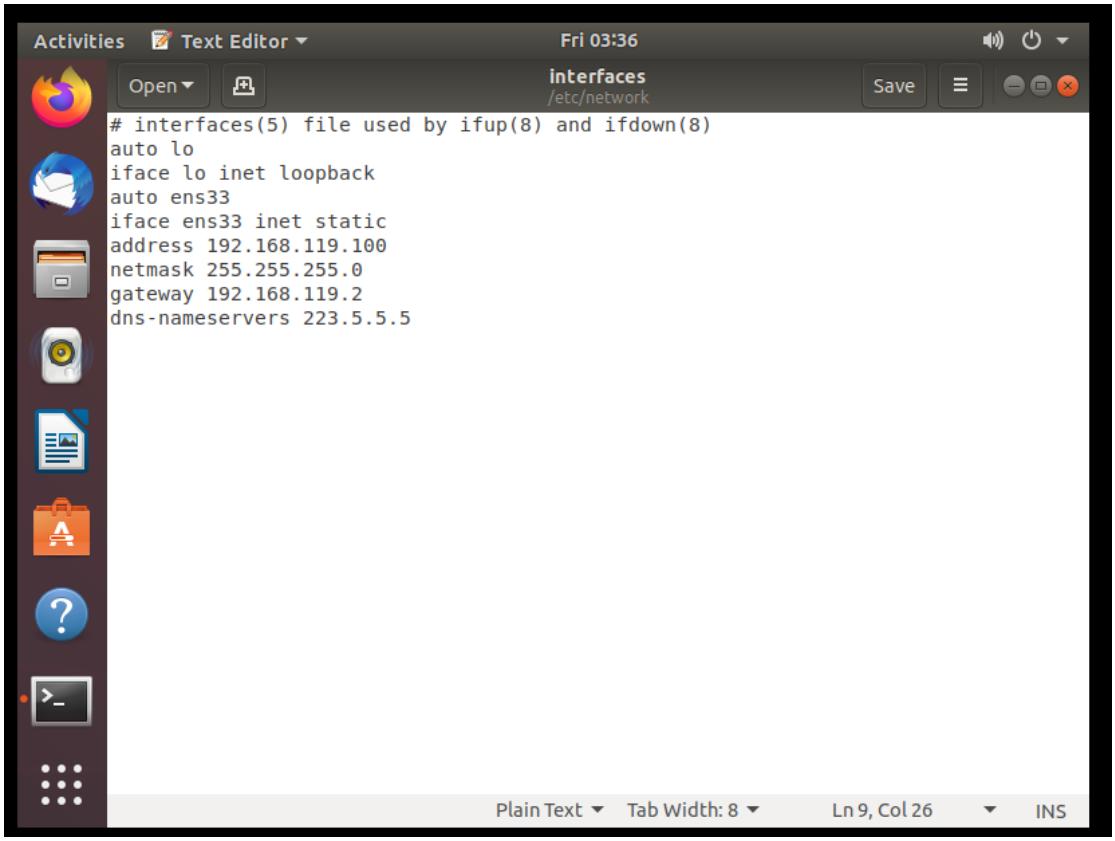
1 虚拟机硬件情况和网络配置信息

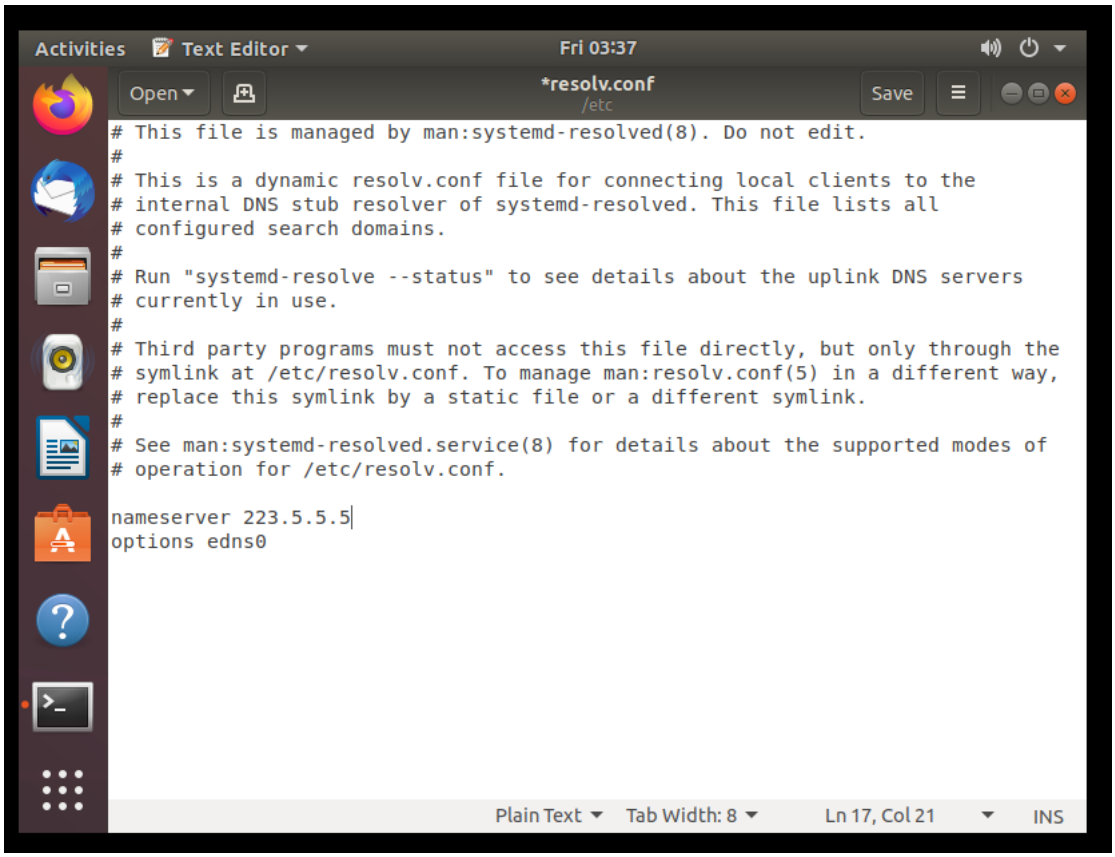
宿主机处理器：Intel® Core™ i5-8250U
宿主机操作系统：Windows 10，64 位
宿主机内存: 8G
虚拟机软件：VMware Workstation Pro 12
虚拟机操作系统：ubuntu 18.04，2G 内存
虚拟机硬件：

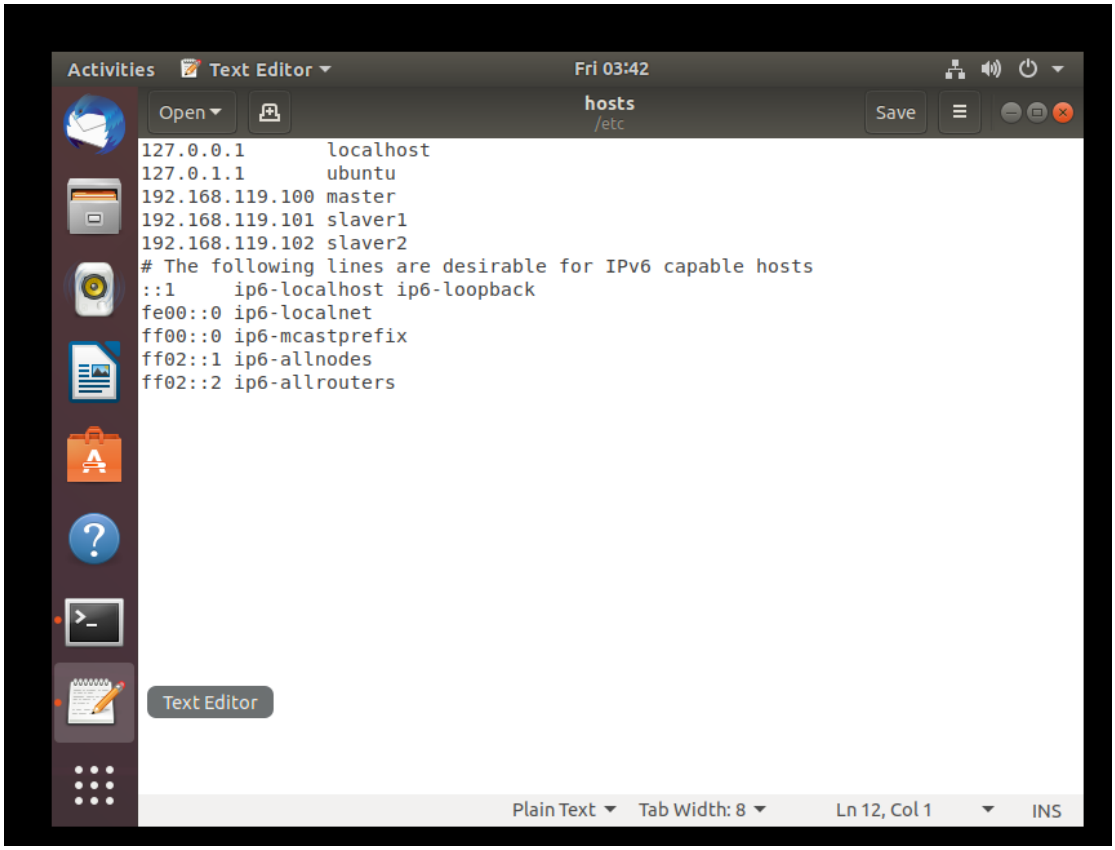
虚拟机网络环境：

序号	IP 地址	主机名	系统	User
1	192.168.119.100	master	Ubuntu18.04	master
2	192.168.119.101	slaver1	Ubuntu18.04	slaver1
3	192.168.119.102	slaver2	Ubuntu18.04	slaver2

master 主机上的网络环境截图：



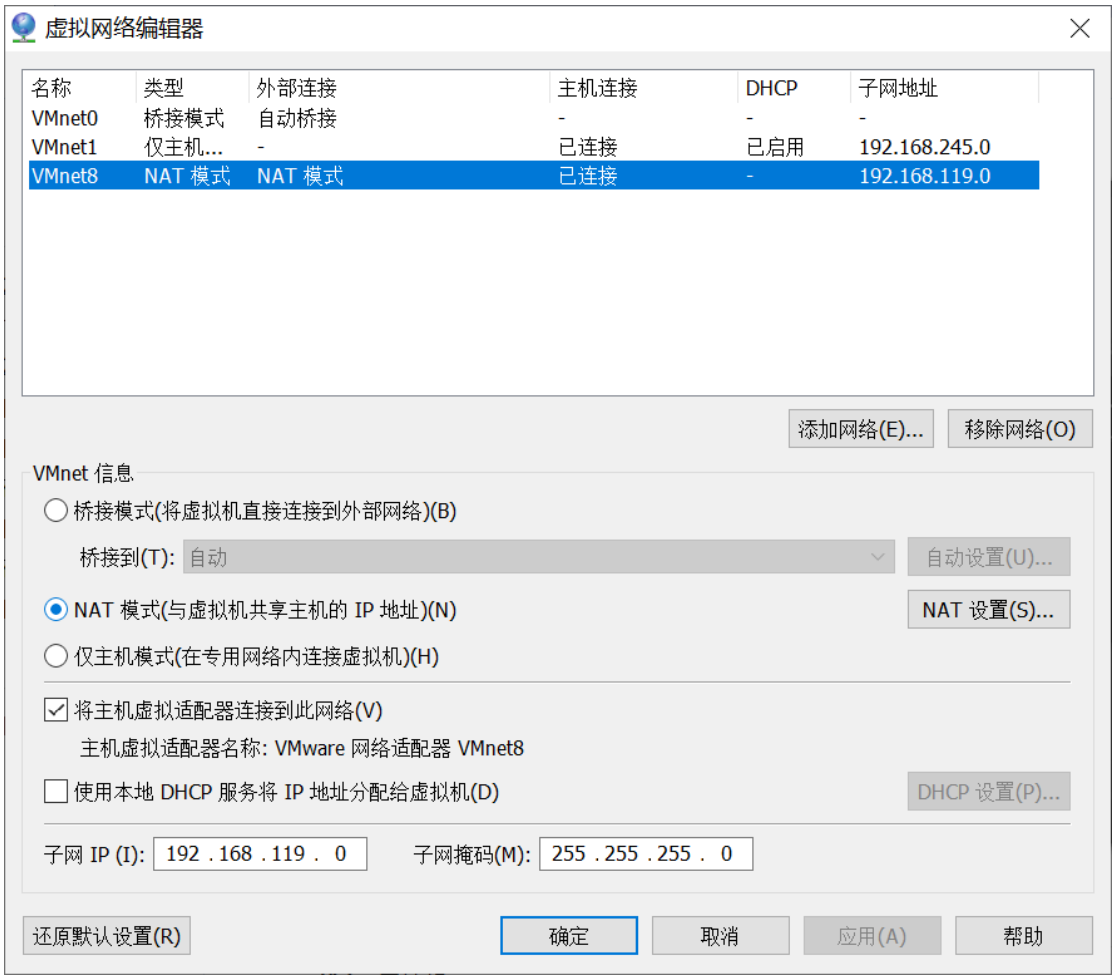

Activities
Text Editor
Fri 03:37
Open
*resolv.conf
/etc
Save
This file is managed by man:systemd-resolved(8). Do not edit.
#
This is a dynamic resolv.conf file for connecting local clients to the
internal DNS stub resolver of systemd-resolved. This file lists all
configured search domains.
#
Run "systemd-resolve --status" to see details about the uplink DNS servers
currently in use.
#
Third party programs must not access this file directly, but only through the
symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
replace this symlink by a static file or a different symlink.
#
See man:systemd-resolved.service(8) for details about the supported modes of
operation for /etc/resolv.conf.
nameserver 223.5.5.5
options edns0
Plain Text
Tab Width: 8
Ln 17, Col 21
INS


Activities
Text Editor
Fri 03:42
Open
hosts
/etc
Save
127.0.0.1 localhost
127.0.1.1 ubuntu
192.168.119.100 master
192.168.119.101 slaver1
192.168.119.102 slaver2
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
Text Editor
Plain Text
Tab Width: 8
Ln 12, Col 1
INS

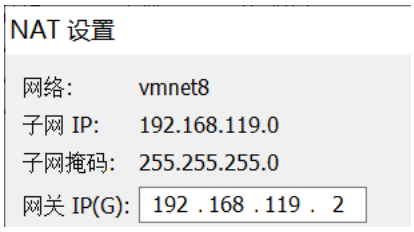
2 集群搭建操作记录

2.1 Vmware 网络配置以及 Ubuntu 虚拟机网络配置

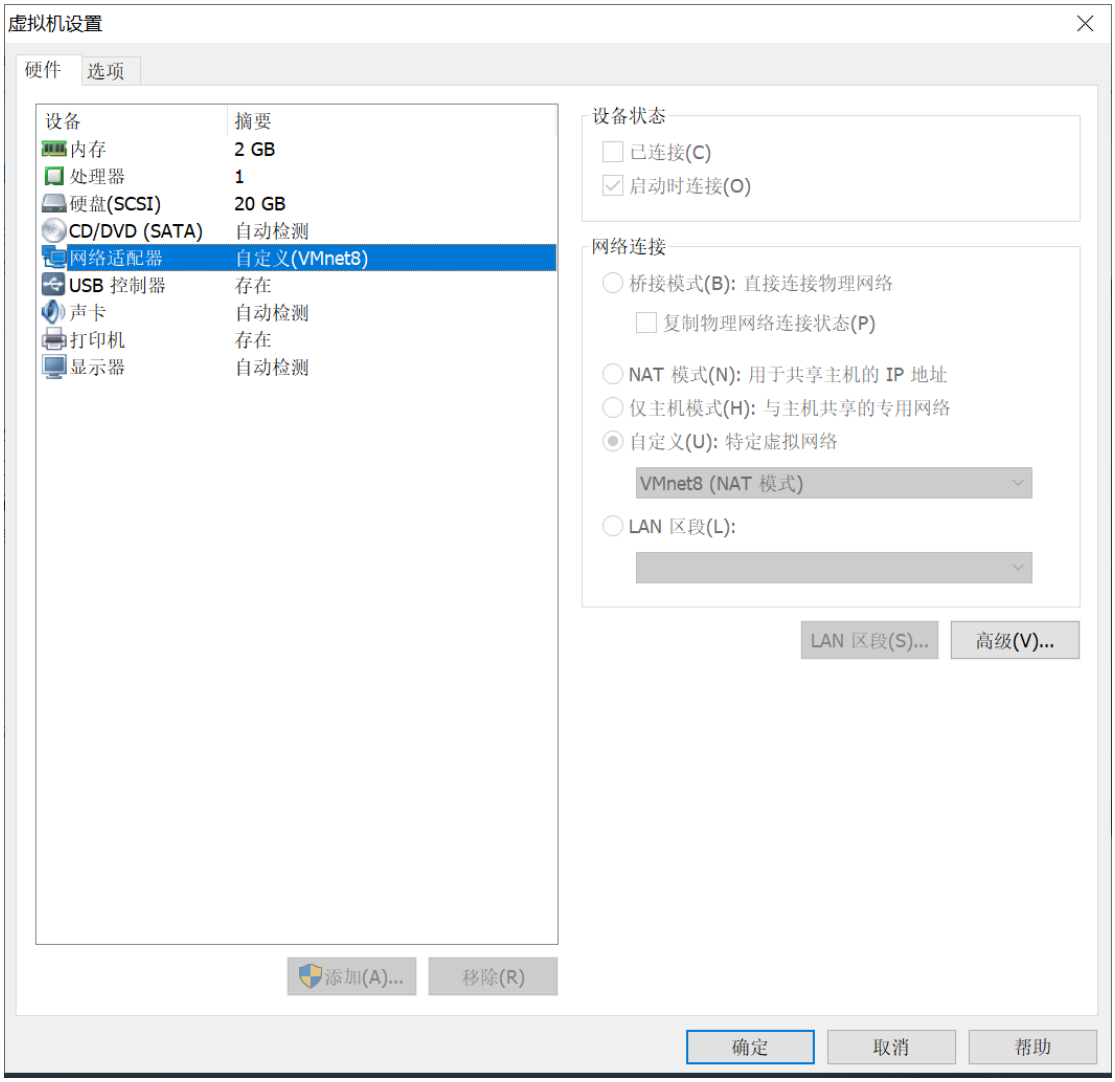
打开 Vmware 中的虚拟网络编辑器，我们需要实现静态 IP 地址，所以在 VMnet8 网络中，去掉勾选“使用本地 DHCP 服务将 IP 地址分配给虚拟机“。子网 IP 为：192.168.119.0，子网掩码为：255.255.255.0。设置三个虚拟机的 IP 地址分别为 192.168.119.100(master)，192.168.119.101(slaver1)，192.168.119.102(slaver2)。



点击 NAT 设置，查看网关为 192.168.119.2。



右键点击自己建立的虚拟机，点击“设置”，查看如下界面，选择网络适配器，确定网络连接选中的是“自定义”中的 VMnet8(NAT 模式)。最后点击确定，开启虚拟机。



打开 Ubuntu 的终端，输入：sudo gedit /etc/network/interfaces，进行如下编辑并保存。

auto lo

iface lo inet loopback

auto ens33

iface ens33 inet static

address 192.168.119.100

netmask 255.255.255.0

gateway 192.168.153.2

dns-nameservers 223.5.5.5

配置 DNS 服务器，在终端中输入：sudo gedit /etc/resolv.conf，进行如下编辑并保存。

nameserver 223.5.5.5

然后，在终端中输入：sudo /etc/init.d/networking restart，重启网络。

2.2 分布式环境搭建

修改 hosts 文件，将 IP 与主机名的映射添加到 hosts 文件中 。在终端中输入：sudo gedit /etc/hosts，进行如下编辑并保存。

192.168.119.100　master

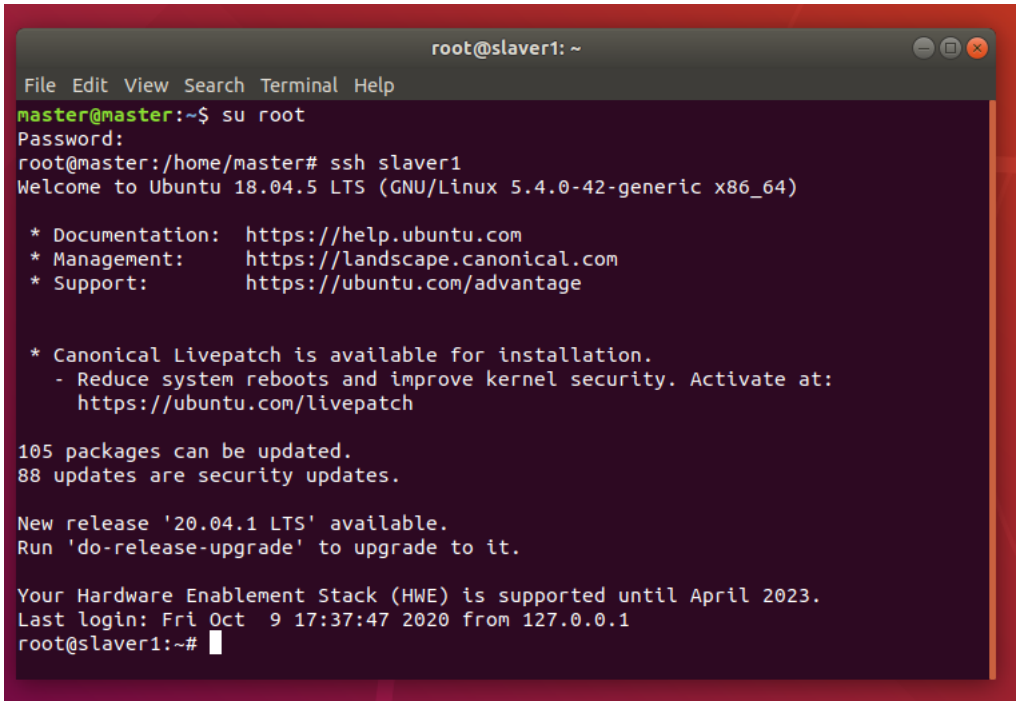
192.168.119.101　slaver1

192.168.119.102　slaver2

修改完成后保存执行如下命令：
source /etc/hosts
关闭虚拟机，在 Vmware 界面对 master 进行克隆，克隆出两个节点 slaver1 和 slaver2。
在每个机器上使用以下命令修改本机名称，分别改成 master，slaver1，slaver2。
sudo gedit /etc/hostname
在每个机器的/etc/network/interfaces 中，将机器的 ip 设置为 static，并分别分配固定 ip。分别为：192.168.119.100，192.168.119.101，192.168.119.102。
对虚拟机进行重启，使相关配置生效。

2.3 免密 ssh 配置

在三台虚拟机中，使用以下命令安装 ssh：
sudo apt-get install openssh-server
在 master 节点上进行 ssh 配置，在 master 节点上执行以下命令。
su root
cd /root/.ssh
ssh-keygen -t rsa
ssh-copy-id -i /root/.ssh/id_rsa.pub master
ssh-copy-id -i /root/.ssh/id_rsa.pub slaver1
ssh-copy-id -i /root/.ssh/id_rsa.pub slaver2
在 master 上对每一个节点进行测试，看是否能进行免密登录：
ssh slaver1



2.4 Java 环境搭建

在/usr 目录下新建 java 文件夹，然后将 jdk 压缩包复制在 java 文件夹下并进行解压。执行：
cd /usr
mkdir java
sudo tar -zxvf jdk-8u201-linux-x64.tar.gz
在/etc/profile 内配置环境变量：gedit /etc/profile。添加如下信息：
export JAVA_HOME=/usr/java/jdk1.8.0_201

export CLASSPATH=:$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
刷新环境配置。然后检测 Java 版本：
source /etc/profile
java -version

2.5 scala 环境配置

将 scala 压缩包复制在 java 文件夹下并进行解压。执行：
sudo tar -zxvf scala-2.11.8.tgz
在/etc/profile 内配置环境变量：gedit /etc/profile。添加如下信息：
export SCALA_HOME=/usr/scala-2.11.8
export PATH=$PATH:$SCALA_HOME/bin
刷新环境配置。然后检测 scala 版本：
source /etc/profile
scala -version

2.6 hadoop 环境配置

解压并移动到/usr/java 目录
sudo tar -zxvf hadoop-2.7.7.tar.gz
修改相应的配置文件。修改/etc/profile，增加如下内容：
export HADOOP_HOME=/usr/java/hadoop-2.7.7
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
在 hadoop-2.7.7 目录下添加目录：
mkdir tmp
mkdir hdfs
mkdir hdfs/name
mkdir hdfs/data
修改$HADOOP_HOME/etc/hadoop/hadoop-env.sh，修改 JAVA_HOME 如下：
export JAVA_HOME=/usr/java/jdk1.8.0_201
修改$HADOOP_HOME/etc/hadoop/slaves，将原来的 localhost 删除，添加如下内容：
slaver1
slaver2
修改$HADOOP_HOME/etc/hadoop/core-site.xml，修改为如下内容：

```
<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>/usr/java/hadoop-2.7.7/tmp</value>
    </property>
```

```
</configuration>
```

修改$HADOOP_HOME/etc/hadoop/hdfs-site.xml。

```
<configuration>
    <property>
      <name>dfs.namenode.secondary.http-address</name>
      <value>Master:50090</value>
    </property>
    <property>
      <name>dfs.replication</name>
      <value>2</value>
    </property>
    <property>
      <name>dfs.namenode.name.dir</name>
      <value>file:/usr/java/hadoop-2.7.7/hdfs/name</value>
    </property>
    <property>
      <name>dfs.datanode.data.dir</name>
      <value>file:/usr/java/hadoop-2.7.7/hdfs/data</value>
    </property>
</configuration>
```

在$HADOOP_HOME/etc/hadoop 目录下复制 template，生成 xml，命令如下：

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

修改$HADOOP_HOME/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.address</name>
        <value>Master:19888</value>
    </property>
</configuration>
```

修改$HADOOP_HOME/etc/hadoop/yarn-site.xml。

```
<configuration>
     <property>
          <name>yarn.nodemanager.aux-services</name>
          <value>mapreduce_shuffle</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>
</configuration>
```

2.7 spark 环境配置

解压并移动到相应目录，命令如下：

sudo tar -zxvf spark-2.4.7-bin-hadoop2.7.tgz

修改/etc/profie，增加如下内容：

export SPARK_HOME=/usr/java/spark-2.4.7-bin-hadoop2.7/

export PATH=$PATH:$SPARK_HOME/bin

在$SPARK_HOME/conf/目录下复制 spark-env.sh.template 成 spark-env.sh

cp spark-env.sh.template spark-env.sh

修改$SPARK_HOME/conf/spark-env.sh，添加如下内容：

export SCALA_HOME=/usr/java/scala-2.11.8

export JAVA_HOME=/usr/java/jdk1.8.0_201

export HADOOP_HOME=/usr/java/hadoop-2.7.7

export SPARK_WORKER_MEMORY=1g

export HADOOP_CONF_DIR=/usr/java/hadoop-2.7.7/etc/Hadoop

在$SPARK_HOME/conf/目录下复制 slaves.template 成 slaves

cp slaves.template slaves

修改$SPARK_HOME/conf/slaves，添加如下内容：

master

slaver1

slaver2

2.8 节点环境配置及测试

将配置好的环境拷贝到 Slaver1 和 Slaver2 节点。

scp -r /usr/java root@slaver1:/usr

scp -r /etc/profile root@slaver1:/etc/profile

scp -r /usr/java root@slaver2:/usr
scp -r /etc/profile root@slaver2:/etc/profile
在每个节点上刷新环境配置: source /etc/profile。
ssh slaver1
source /etc/profile
exit
ssh slaver2
source /etc/profile
exit
在 master 节点启动 Hadoop，启动之前格式化一下 namenode：
hadoop namenode -format
启动：
/usr/java/hadoop-2.7.7/sbin/start-all.sh
查看 Hadoop 是否启动成功，输入命令：jps
Master 显示：SecondaryNameNode，ResourceManager，NameNode。
Slaver 显示：NodeManager，DataNode。

```
root@master:/home/master# jps
6993 NameNode
7235 SecondaryNameNode
7731 Master
7380 ResourceManager
11373 Jps
7854 Worker
```

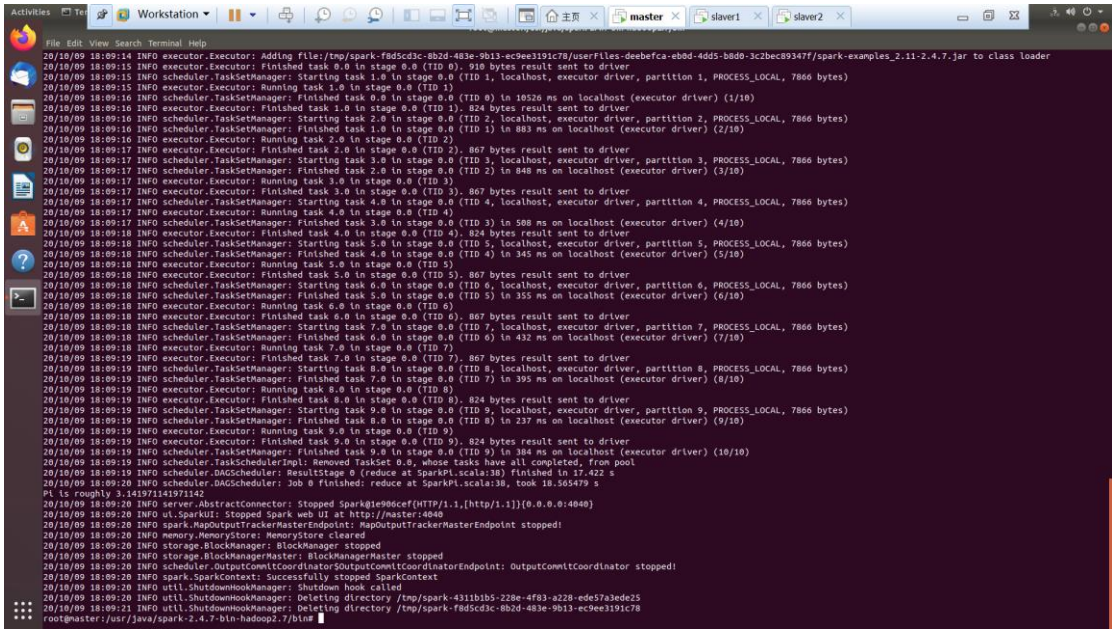
```
root@slaver1:/usr/java/hadoop-2.7.7# jps
3475 NodeManager
3316 DataNode
5243 Jps
3695 Worker
```

在 master 节点启动 Spark：/usr/java/spark-2.4.7-bin-hadoop2.7/sbin/start-all.sh
查看 Spark 是否启动成功，输入命令：jps
Master 在 Hadoop 的基础上新增了：Master。
Slaver 在 Hadoop 的基础上新增了：Worker。

```
root@master:/home/master# jps
6993 NameNode
7235 SecondaryNameNode
7731 Master
7380 ResourceManager
11373 Jps
7854 Worker
```

```
root@slaver1:/usr/java/hadoop-2.7.7# jps
3475 NodeManager
3316 DataNode
5243 Jps
3695 Worker
```

另外我们根据官网，运行一下代码：
/usr/java/spark-2.4.7-bin-hadoop2.7/bin/run-example SparkPi 10
结果如下，计算出 pi：



在输入以下代码后，可以在 WebUI 看到运行应用：

$SPARK_HOME/bin/spark-submit --class org.apache.spark.examples.SparkPi --master spark://master:7077 /usr/java/spark-2.4.7-bin-hadoop2.7/examples/jars/spark-examples_2.11-2.4.7.jar