

Hadoop完全分布式安装

本节介绍安装Hadoop完全分布式的主要步骤。本次实验使用三个节点来搭建集群环境：一台机器作为Master节点，具体的集群规划如下表所示：

主机名	master	slaves1	slaves2
HDFS	NameNode DataNode	DataNode SecondaryNameNode	DataNode
YARN	NodeManagerr ResourceManager	NodeManager	NodeManage

在安装Hadoop完全分布式前，我们需要完成一些准备工作，包括网络配置、关闭防火墙、安装SSH和安装JDK环境等。

一、网络配置：

动态分配的IP地址是临时的，它会在一定时间内释放该IP地址供其他机器使用，因此使用该方式获取的IP地址不是固定的，这样会导致集群不稳定。因此我们需要设置为静态IP，具体操作如下：

- 1. 打开【/etc/sysconfig/network-scripts/ifcfg-ens33】文件。

```
1 | vi /etc/sysconfig/network-scripts/ifcfg-ens33
```

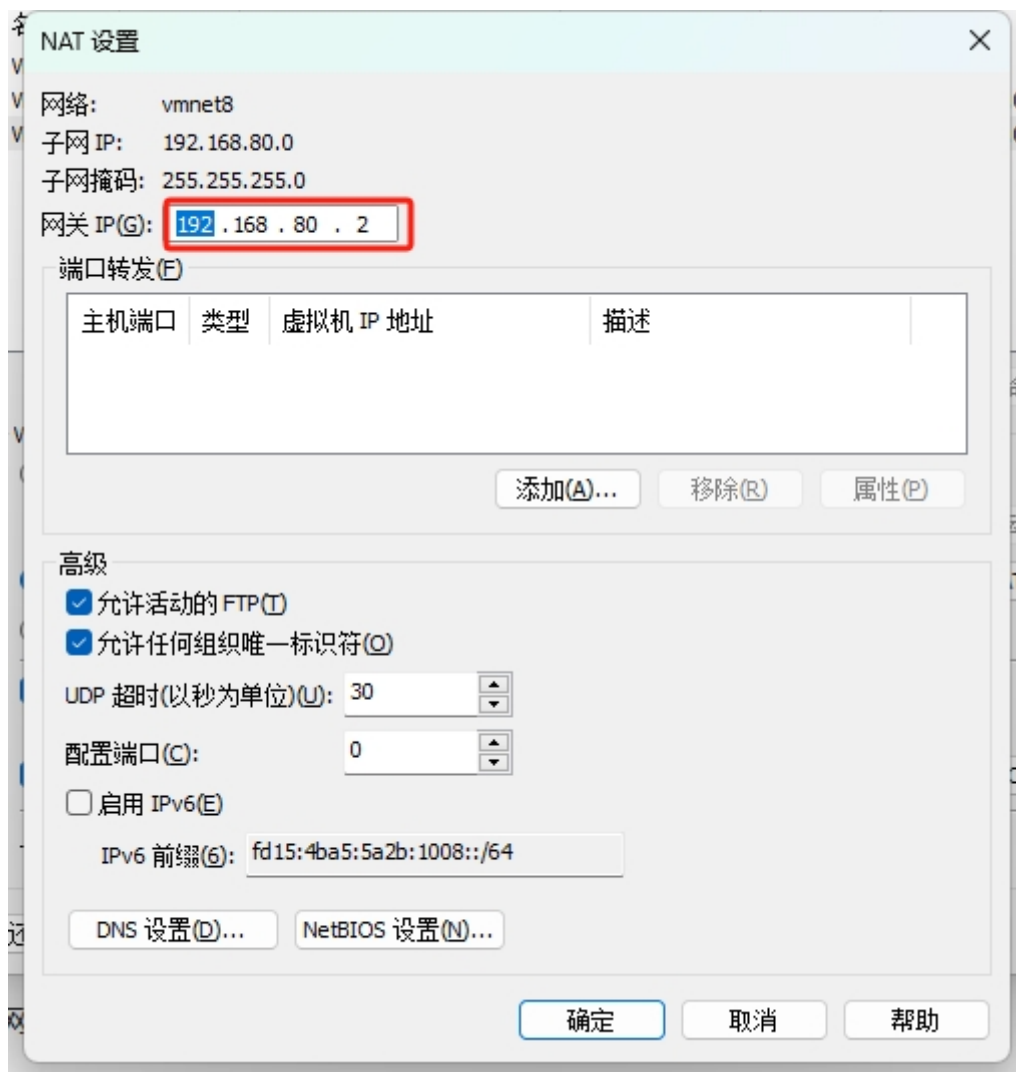
- 2. 该文件为网卡配置文件，对该文件做以下修改：

```
1 | BOOTPROTO="static"
```

打开虚拟机【编辑】->【虚拟网络编辑器】，查看VMnet8网卡的子网IP地址。



点击【NAT设置】界面，查看网关IP地址。



再向ifcfg-ens33添加以下内容：

- 1 IPADDR=192.168.80.160 #前3个数必须与网关IP的前3个数必须一致，最后一个数则随机填入1-255的数字，组成新的IP地址，该IP必须是未被占用的。
- 2 GATEWAY=192.168.80.2 #输入VMnet8网关IP地址
- 3 DNS1=114.114.114.114

注意：此处每台机器节点配置不一样，大家先进行子网IP查询，再添加以上内容。

3. 重启网络

- 1 `systemctl restart network`

4. 网络测试，尝试连接百度地址，如果能正常连接成功，则表示网络配置完成。

- 1 `ping www.baidu.com`

```
[root@master ~]# ping www.baidu.com
PING www.baidu.com (120.232.145.185) 56(84) bytes of data.
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=1 ttl=128 time=13.6 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=2 ttl=128 time=13.5 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=3 ttl=128 time=14.3 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=4 ttl=128 time=14.2 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=5 ttl=128 time=14.1 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=6 ttl=128 time=15.0 ms
64 bytes from 120.232.145.185 (120.232.145.185): icmp_seq=7 ttl=128 time=13.6 ms
```

二、关闭防火墙和SELinux

注意：这个步骤只是为了我们学习方便，在实际工作中绝对不可以关闭防火墙，只能在防火墙配置需要开放的端口即可。

1. 查看当前防火墙状态

```
1 | systemctl status firewalld.service
```

CentOS 7默认是开机启动防火墙。因此我们要关闭防火墙并停止开启自动启动防火墙设置。

```
[root@master ~]# systemctl status firewalld.service
■ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-12-12 16:10:45 CST; 1h 27min ago
     Docs: man:firewalld(1)
   Main PID: 765 (firewalld)
    CGroup: /system.slice/firewalld.service
            └─765 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid
```

2. 关闭当前防火墙

```
1 | systemctl stop firewalld.service
```

3. 关闭防火墙的开机自启

```
1 | systemctl disable firewalld
```

4. 关闭后再查看防火墙状态。

```
1 | systemctl status firewalld.service
```

```
[root@master ~]# systemctl status firewalld.service
■ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
```

如果防火墙状态变为Inactive(dead)状态，则表示防火墙关闭成功。

关闭完防火墙后，我们即可使用XShell工具连接。

5. Linux有一个安全模块：SELinux，用以限制用户和程序的相关权限，来确保系统的安全稳定。SELinux的配置同防火墙一样，非常复杂，课程中不多涉及，后续视情况可以出一章SELinux的配置课程。在当前，我们只需要关闭SELinux功能，避免导致后面的软件运行出现问题即可，

修改/etc/sysconfig/selinux文件

```
1 | vim /etc/sysconfig/selinux
```

修改文件内容。

```
1 | #将第7行的SELINUX=enforcing改为,保存退出后，重启虚拟机即可，千万要注意
   disabled单词不要写错，不然无法启动系统
2 | SELINUX=disabled
```

三、安装常用工具

1. 安装net-tools工具

```
1 | yum install -y net-tools
```

2. 安装vim工具

```
1 | yum install -y vim
```

四、安装JDK1.8环境

后续的大数据集群软件，多数是需要Java运行环境的，所以我们为==每一台==机器都配置JDK环境。

1. 从Java官网中下载JDK1.8安装包，下载地址为：[Java Downloads | Oracle](https://www.oracle.com/technetwork/java/javase-downloads-1344635.html)

Java SE 8U391 Downloads and Oracle Cloud registration

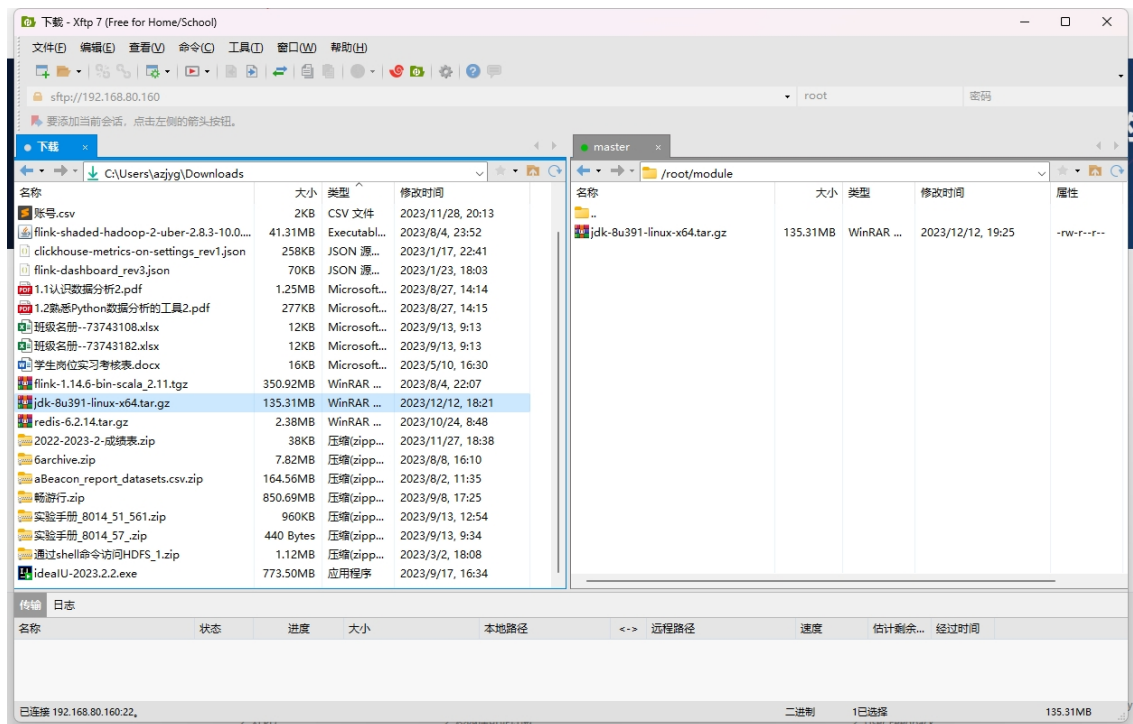
Linux	macOS	Solaris	Windows
Product/file description	File size		Download
ARM64 RPM Package	71.06 MB		jdk-8u391-linux-aarch64.rpm
ARM64 Compressed Archive	71.23 MB		jdk-8u391-linux-aarch64.tar.gz
x86 RPM Package	140.62 MB		jdk-8u391-linux-i586.rpm
x86 Compressed Archive	138.69 MB		jdk-8u391-linux-i586.tar.gz
x64 RPM Package	137.36 MB		jdk-8u391-linux-x64.rpm
x64 Compressed Archive	135.33 MB		jdk-8u391-linux-x64.tar.gz

2. 上传JDK安装包到CentOS虚拟机。

创建一个目录/root/module，专门存放软件安装包

```
1 | mkdir /root/module
```

使用XFTP上传JDK安装包到/root/module目录下。



3. 解压DK安装包。

```
1 cd /root/module
2 tar -zxvf jdk-8u391-linux-x64.tar.gz -C /usr/local/
```

4. 更改jdk目录名字

```
1 cd /usr/local/
2 mv jdk1.8.0_391 jdk1.8
```

5. 配置环境变量，编辑/etc/profile文件，将JDK路径添加到PATH环境变量中。

```
1 vim /etc/profile
```

在/etc/profile文件末端添加以下内容

```
1 #JAVA HOME
2 export JAVA_HOME=/usr/local/jdk1.8
3 export PATH=$PATH:$JAVA_HOME/bin
```

6. 刷新环境变量

```
1 source /etc/profile
```

7. 验证JDK是否生效

```
1 java -version
```

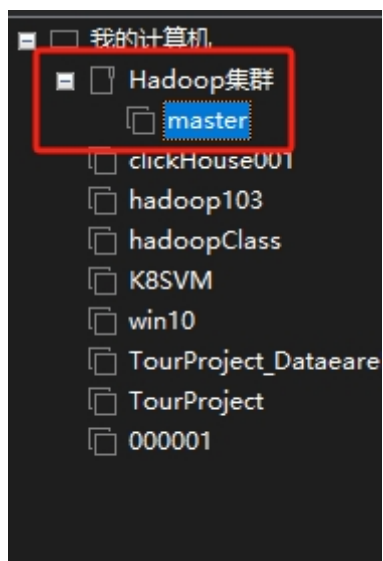
```
[root@master local]# java -version
java version "1.8.0_391"
Java(TM) SE Runtime Environment (build 1.8.0_391-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.391-b13, mixed mode)
```

出现Java版本号信息，则证明JDK安装完成。

五、复制CentOS镜像

安装集群化软件，首要条件就是要有多台Linux服务器可用。我们可以使用VMware提供的克隆功能，将我们的虚拟机额外克隆出3台来使用。

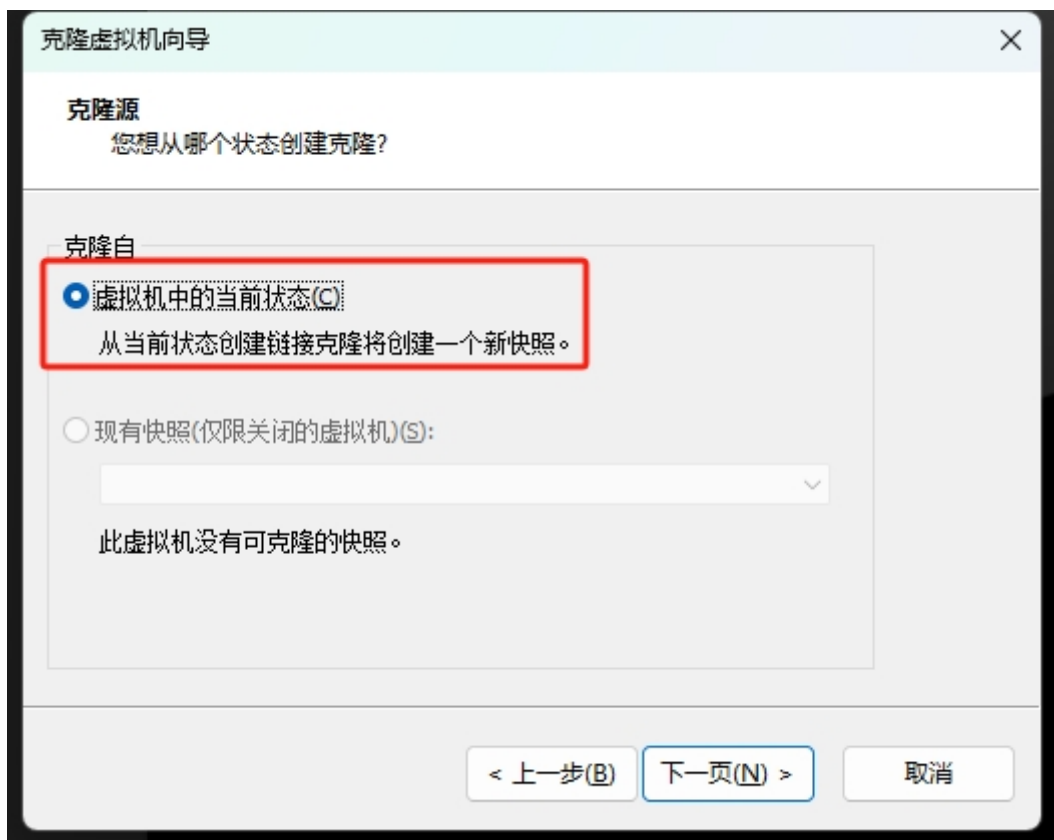
1. 首先，关闭当前CentOS系统虚拟机
2. 在VMWare工具中新建文件夹，命名为：`Hadoop集群`，将master机器拖到Hadoop集群文件夹中。



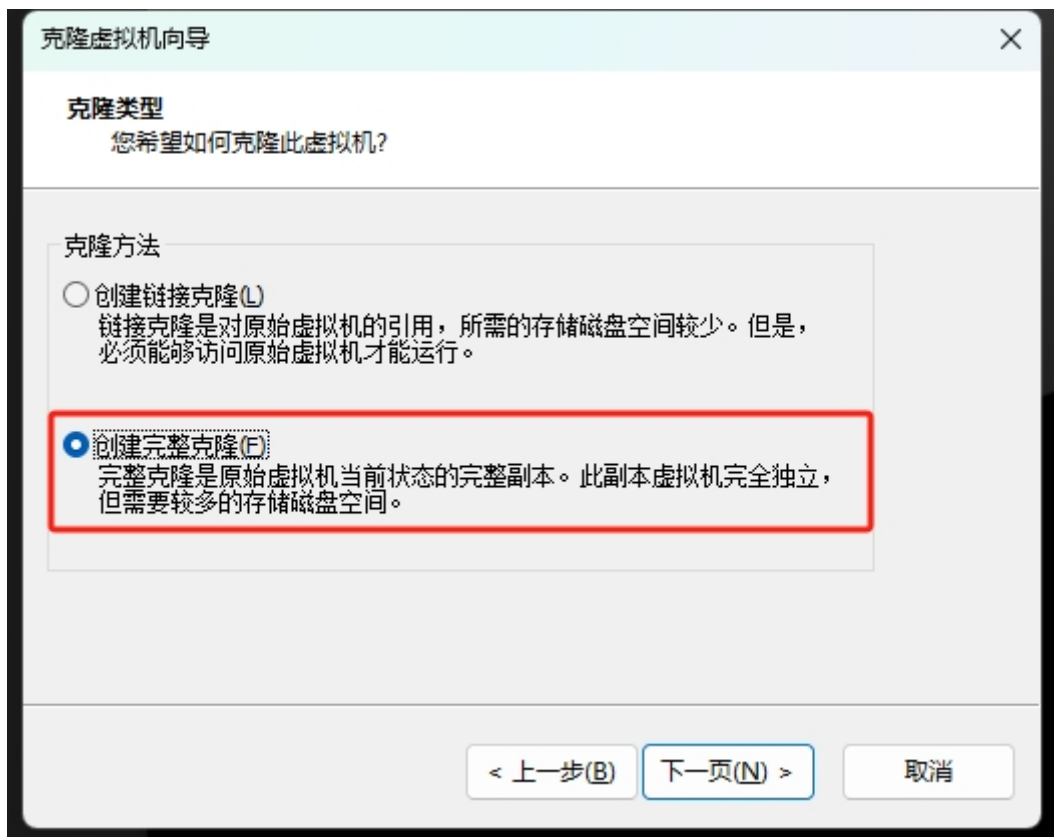
3. 克隆master节点。鼠标右键点击【master】节点，选择【管理】->【克隆】进入虚拟机克隆管理界面。



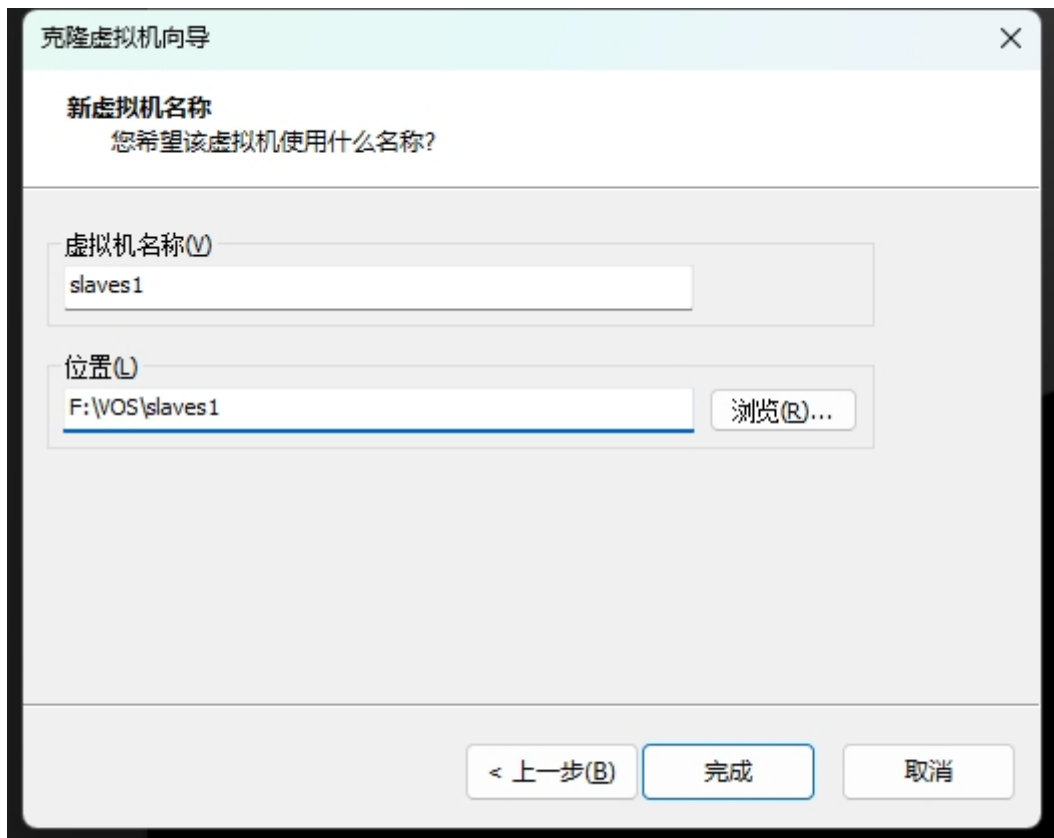
克隆虚拟机当前状态。



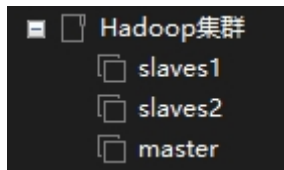
选择【创建完整克隆】



虚拟机名称为salves1,保存路径设置到不还原盘中。点击【完成】。



4. 按照以上操作，克隆出第二个节点：slaves2,并将slaves1与slaves2放到 Hadoop集群 文件夹中。



5. 开启slaves1节点，修改slaves1节点的配置。

① 修改slaves1节点的主机名为slaves1

```
1 hostnamectl set-hostname slaves1
```

② 修改slaves1节点的固定ip为：192.168.80.161（注意IP地址前3个数需跟VMnet8 IP网关地址前3个数必须一致）

```
1 vim /etc/sysconfig/network-scripts/ifcfg-ens33
```

修改IPADDR的值

```
1 IPADDR=192.168.80.161
```

6. 开启slaves2节点，修改slaves2节点的配置。

按照第5步步骤，将slaves2节点的主机名设置为slaves2，ip设置为：192.168.80.162（注意IP地址前3个数需跟VMnet8 IP网关地址前3个数必须一致）

7. 配置主机名映射

修改CentOS系统中的/etc/hosts文件。

```
1 vim /etc/hosts
```

在文件末端添加以下内容

```
1 192.168.80.160 master
2 192.168.80.161 slaves1
3 192.168.80.162 slaves2
```

注意：3台节点都需要修改

六、配置SSH免密登录

后续安装的集群，都需要远程登录以及远程执行命令，我们可以简单起见，配置三台Linux服务器之间的免密码互相SSH登陆。

1. 在每一台机器执行

```
1 ssh-keygen -t rsa
```

一直回车确认即可。

2. 在每一台机器节点执行一下指令，




```
1 ssh-copy-id master
2 ssh-copy-id slaves1
3 ssh-copy-id slaves2
4 #进行免密操作时会询问是否继续连接，输入“yes”后，再输入登录密码完成操作。
```

3. 执行完毕后，使用SSH测试远程登录，如果三台机器能相互登录，且无须输入密码，即代表免密配置成功。

```
1 ssh master      #远程登录master节点
2 exit            #退出登录
3 ssh slaves1     #远程登录slaves1节点
4 exit            #退出登录
5 ssh slaves2     #远程登录slaves2节点
6 exit            #退出登录
```

七、Hadoop集群部署

1. 下载Hadoop安装包。打开浏览器，在浏览器地址栏中输入以下地址：<https://archive.apache.org/dist/hadoop/common/hadoop-3.1.3/>。
2. 选择hadoop-3.1.3.tar.gz 版本下载hadoop安装包。

	hadoop-3.1.3-src.tar.gz.asc	2020-07-03 04:36	473
	hadoop-3.1.3-src.tar.gz.sha512	2020-07-03 04:36	195
	hadoop-3.1.3.tar.gz	2020-07-03 04:37	322M
	hadoop-3.1.3.tar.gz.asc	2020-07-03 04:37	473
	hadoop-3.1.3.tar.gz.sha512	2020-07-03 04:36	191

下载完毕后，使用XFTP工具将安装包上传到master节点的/root/module目录下。

3. 解压hadoop安装包

```
1 cd /root/module
2 #解压Hadoop压缩包
3 tar -zxvf hadoop-3.1.3.tar.gz -C /usr/local/
4
5 #更改hadoop目录名称
6 mv /usr/local/hadoop-3.1.3 /usr/local/hadoop
```

4. 配置Hadoop环境变量

```
1 vim /etc/profile
```

在/etc/profile文件末端添加Hadoop路径到环境变量中。

```
1 #HADOOP ENV
2 export HADOOP_HOME=/usr/local/hadoop
3 export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
4 export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
5 export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

保存退出，刷新环境变量。

```
1 source /etc/profile
```

5. 修改Hadoop配置文件。

Hadoop的配置文件要修改的地方很多，请细心！！！！进入到/usr/local/hadoop/etc/hadoop文件夹中，配置文件都在这里

```
1 cd /usr/local/hadoop/etc/hadoop
```

① 修改hadoop-env.sh

```
1 vim hadoop-env.sh
```

在文件开头添加以下内容

```
1 export JAVA_HOME=/usr/local/jdk1.8
2
3 export HDFS_NAMENODE_USER=root
4 export HDFS_DATANODE_USER=root
5 export HDFS_SECONDARYNAMENODE_USER=root
6 export YARN_RESOURCEMANAGER_USER=root
7 export YARN_NODEMANAGER_USER=root
8 export YARN_PROXYSERVER_USER=root
```

② 修改core-site.xml

```
1 vim core-site.xml
```

在 <configuration> 标签之间添加fs.defaultFS属性，具体配置如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8 http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
13 implied.
14 See the License for the specific language governing permissions and
15 limitations under the License. See accompanying LICENSE file.
16 -->
17 <!-- Put site-specific property overrides in this file. -->
18 <configuration>
19   <property>
20     <!-- 指定NameNode的地址 -->
21     <property>
22       <name>fs.defaultFS</name>
23       <value>hdfs://master:9000</value>
24     </property>
25
26     <!-- 指定hadoop数据的存储目录 -->
27     <property>
28       <name>hadoop.tmp.dir</name>
29       <value>/usr/local/hadoop/tmp</value>
30     </property>
31 </configuration>
```

③ 修改hdfs-site.xml文件

```
1 vim hdfs-site.xml
```

具体配置内容如下所示：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8 http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
```

```

12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
    implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20     <!-- 2nn web端访问地址-->
21     <property>
22         <name>dfs.namenode.secondary.http-address</name>
23         <value>slaves1:9890</value>
24     </property>
25     <!-- namenode 数据存放地址-->
26     <property>
27         <name>dfs.namenode.name.dir</name>
28         <value>file:///usr/local/hadoop/tmp/hdfs/name</value>
29     </property>
30     <!-- datanode 数据存放地址-->
31     <property>
32         <name>dfs.datanode.data.dir</name>
33         <value>file:///usr/local/hadoop/tmp/hdfs/data</value>
34     </property>
35     <!-- 设置副本数量 -->
36     <property>
37         <name>dfs.replication</name>
38         <value>3</value>
39     </property>
40 </configuration>

```

④ 修改mapred-site.xml文件

```
1 vim mapred-site.xml
```

具体配置内容如下：

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8 http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
    implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->

```

```

16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20     <!-- 指定MapReduce程序运行在Yarn上 -->
21     <property>
22         <name>mapreduce.framework.name</name>
23         <value>yarn</value>
24     </property>
25
26     <property>
27         <name>mapreduce.jobhistory.address</name>
28         <value>master:10020</value>
29     </property>
30     <property>
31         <name>mapreduce.jobhistory.webapp.address</name>
32         <value>master:19888</value>
33     </property>
34     <property>
35         <name>yarn.app.mapreduce.am.env</name>
36         <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
37     </property>
38     <property>
39         <name>mapreduce.map.env</name>
40         <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
41     </property>
42     <property>
43         <name>mapreduce.reduce.env</name>
44         <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
45     </property>
46 </configuration>

```

⑤ 修改yarn-site.xml文件

```
1 vim yarn-site.xml
```

具体配置文件如下所示：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4     Licensed under the Apache License, Version 2.0 (the "License");
5     you may not use this file except in compliance with the License.
6     You may obtain a copy of the License at
7         http://www.apache.org/licenses/LICENSE-2.0
8     Unless required by applicable law or agreed to in writing, software
9     distributed under the License is distributed on an "AS IS" BASIS,
10    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
    implied.
11    See the License for the specific language governing permissions and
12    limitations under the License. See accompanying LICENSE file.
13 -->
14 <configuration>

```

```

15      <!-- 指定MR走shuffle -->
16      <property>
17          <name>yarn.nodemanager.aux-services</name>
18          <value>mapreduce_shuffle</value>
19      </property>
20      <!-- 指定ResourceManager的地址-->
21      <property>
22          <name>yarn.resourcemanager.hostname</name>
23          <value>master</value>
24      </property>
25      <!--指定Yarn调度器-->
26      <property>
27          <name>yarn.resourcemanager.scheduler.class</name>
28
29          <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.Fair
30          Scheduler</value>
31      </property>
32  </configuration>

```

⑥修改workers文件

```
1 vim workers
```

在workers文件添加以下内容

```

1 master
2 slaves1
3 slaves2

```

6. 分发hadoop文件夹到slaves1与slaves2节点

```

1 #在master节点执行
2 cd /usr/local
3 #复制hadoop到slaves1与slaves2
4 scp -r hadoop slaves1:/usr/local/
5 scp -r hadoop slaves2:/usr/local/
6 #复制/etc/profile文件到slaves1与slaves2
7 scp /etc/profile slaves1:/etc/profile
8 scp /etc/profile slaves2:/etc/profile

```

7. 格式化NameNode，在master节点上运行以下命令，完成格式化

```
1 hdfs namenode -format
```

当出现【successfully formatted】的时候，代表格式化成功。

```

INFO common.Storage: Storage directory /usr/local/hadoop/tmp/hdfs/name has been successfully formatted.
INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/hdfs/name/current/fsimage.ckpt_000000000000000000 using no compression
INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/hdfs/name/current/fsimage.ckpt_000000000000000000 of size 388 bytes saved in 0 seconds
INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
INFO namenode.FSImageSaver: clean checkpoint: txid = 0 when meet shutdown.
INFO namenode.NameNode: SHUTDOWN_MSG:

```

8. 启动hadoop集群，在master节点执行即可

```
1 | start-all.sh
```

9. 使用jps指令查看进程

master: 在master节点运行着 **NameNode**、**DataNode**、**NodeManager**、**ResourceManager** 4个进程。

```
[root@master hadoop]# jps
4257 DataNode
4099 NameNode
4823 NodeManager
5175 Jps
4667 ResourceManager
[root@master hadoop]#
```

slaves1: 在slaves1节点运行着 **SecondaryNameNode**、**DataNode**、**NodeManager** 3个进程。

```
[root@slaves1 ~]# jps
2498 SecondaryNameNode
2387 DataNode
2707 Jps
2583 NodeManager
[root@slaves1 ~]#
```

slaves2: 在slaves2节点运行着 **NodeManger**、**DataNode** 2个进程。

```
[root@slaves2 ~]# jps
3120 NodeManager
3004 DataNode
3244 Jps
```

10. MapReduce单词统计

① 准备一个words.txt文本。

```
1 | mkdir /root/data
2 | cd /root/data
3 | vim words.txt
```

在words.txt文本中输入以下内容：

```
1 | I Love Hadoop
2 | I Love Spark
3 | Hadoop is good
4 | Spark is fast
```

② 将文本上传到HDFS /data 目录下。

```
1 | hdfs dfs -mkdir /data
2 | hdfs dfs -put /root/data/words.txt /data
```

③ 运行wordcount任务

```
1 | hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar wordcount /data/words.txt /out
```

④ 查看运行结果

```
1 | hdfs dfs -cat /out/part-r-00000
```



```
[root@master data]# hdfs dfs -cat /out/part-r-00000
2023-12-15 10:45:45,419 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
Hadoop 2
I 2
Love 2
Spark 2
fast 1
good 1
is 2
[root@master data]#
```