集群的搭建 -- 对于安徽省职业技能大赛（大数据应用开发赛项）第一模块 hadoop搭建的解析

2024年10月27日 14:10

1.基础环境

Vmware Workstation虚拟机版本：

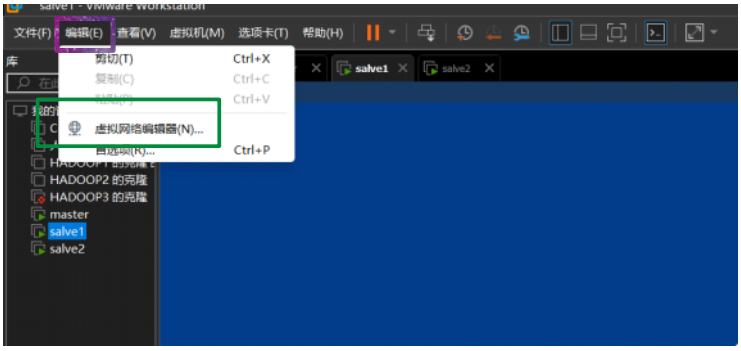
centos7地址规划 如果题目不说的话

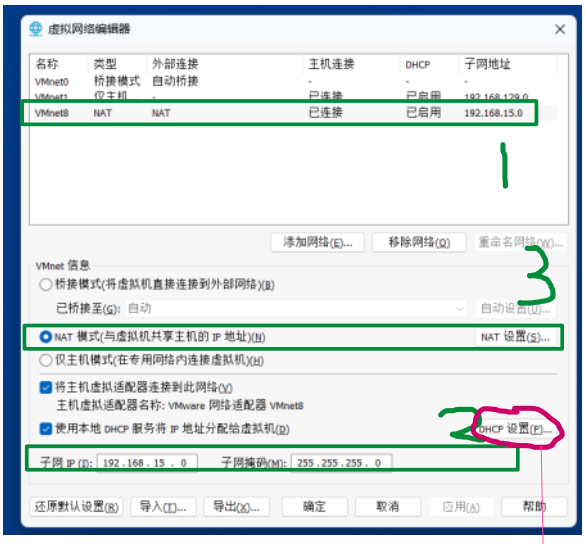
从第三页开始，

主机名	IP	username	password
master	192.168.100.100	xurany	060920
Slave1	192.168.100.101	xurany	060920
Slave2	192.168.100.102	xurany	060920

1).在Vmware workstation 中可能会遇到的问题

2).如果选择了Nat的网卡，在编辑选项需要设置基本的NAT信息



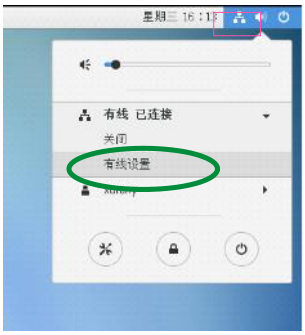


顺便还能解决一下 xshell 租用时间过短的问题

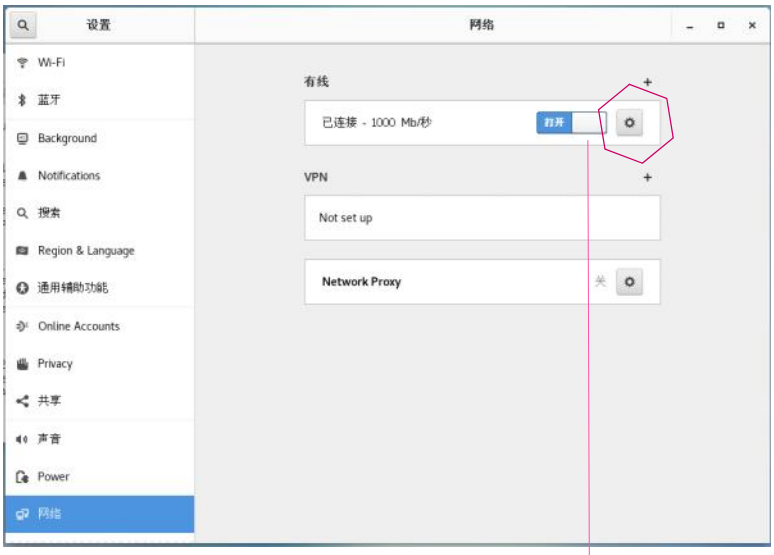
点击上图的DHCP设置

把默认租用时间和最长租用时间改了就行

现在我们来改ip



在右上角，



改成自己需要的就行

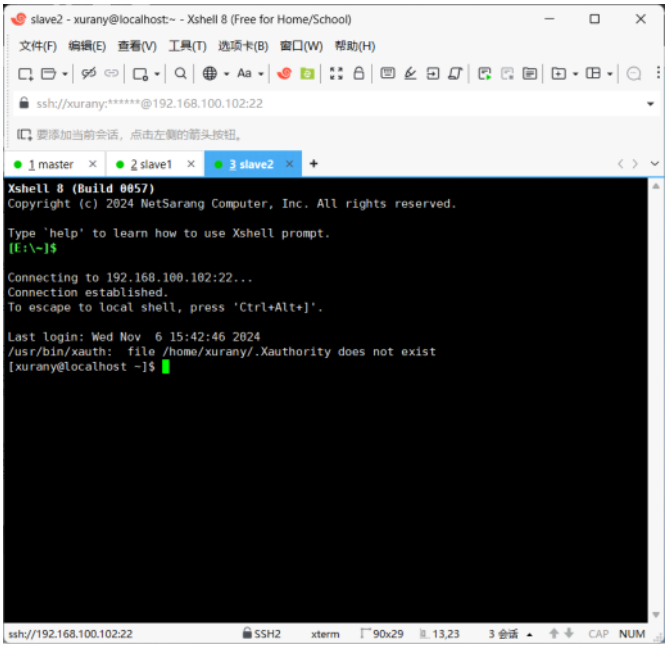
网关的话xxx.xxx.xxx.2(后面这个数不要设置成0)

记得改完刷新一下（网络重新连接一下）

比赛的时侯注意文档，网关类型，IP地址

在master slave1 slave2中改成题目所需要的ip地址

接下来在xshell里配置连接三台虚拟机



全部成功

接下来上传文件

在master节点创建 /opt/softwares目录

命令：mkdir /opt/softwares

```
[root@localhost xurany]# mkdir /opt/softwares
[root@localhost xurany]# ls /opt/
rh  softwares
[root@localhost xurany]#
```

用rz命令将文件夹上传到/opt/softwares目录下

```
[root@localhost softwares]# ls
hadoop-3.1.3.tar.gz  jdk-8u171-linux-x64.tar.gz
[root@localhost softwares]#
```

2.集群基础设置

1.修改每台主机的主机名

master：

命令：hostnamectl set-hostname master

bash

注释：bash是刷新的意思

```
hadoop-3.1.3.tar.gz  jdk-8u171-linux-x64.tar.gz
[root@localhost softwares]# hostnamectl set-hostname master
[root@localhost softwares]# bash
bash
[root@master softwares]# cd /home/xurany/
[root@master xurany]#
```

Slave1：

命令：hostnamectl set-hostname slave1

```
[xurany@localhost ~]$ su
密码:
[root@localhost xurany]# hostnamectl set-hostname slave1
[root@localhost xurany]# bash
[root@slave1 xurany]#
```

Slave2：

命令：hostnamectl set-hostname slave2

```
[xurany@localhost ~]$ hostnamectl set-hostname slave2
==== AUTHENTICATING FOR org.freedesktop.hostname1.set-static-hostname ===
Authentication is required to set the statically configured local host name, as well as th
e pretty host name.
Authenticating as: root
Password:
==== AUTHENTICATION COMPLETE ===
[xurany@localhost ~]$ bash
[xurany@slave2 ~]$ cd /home/xurany/
[xurany@slave2 ~]$ su
密码:
[root@slave2 xurany]#
```

2.修改主机名和IP的映射

命令：vim /etc/hosts

```
[root@master xurany]# vim /etc/hosts
```

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.100.100 master
192.168.100.101 slave1
192.168.100.102 slave2
~
~
~
```

Ping 一下 master slave1 slave2 看设置的是否有问题：

命令：ping master

```
[root@master xurany]# ping master
PING master (192.168.100.100) 56(84) bytes of data.
64 bytes from master (192.168.100.100): icmp_seq=1 ttl=64 time=0.341 ms
64 bytes from master (192.168.100.100): icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from master (192.168.100.100): icmp_seq=3 ttl=64 time=0.104 ms
64 bytes from master (192.168.100.100): icmp_seq=4 ttl=64 time=0.124 ms
64 bytes from master (192.168.100.100): icmp_seq=5 ttl=64 time=0.085 ms
```

命令：ping slave1

```
[root@master xurany]# ping slave1
PING slave1 (192.168.100.101) 56(84) bytes of data.
64 bytes from slave1 (192.168.100.101): icmp_seq=1 ttl=64 time=11.5 ms
64 bytes from slave1 (192.168.100.101): icmp_seq=2 ttl=64 time=1.60 ms
64 bytes from slave1 (192.168.100.101): icmp_seq=3 ttl=64 time=1.09 ms
64 bytes from slave1 (192.168.100.101): icmp_seq=4 ttl=64 time=1.30 ms
64 bytes from slave1 (192.168.100.101): icmp_seq=5 ttl=64 time=5.19 ms
```

命令：ping slave2

```
[root@master xurany]# ping slave2
PING slave2 (192.168.100.102) 56(84) bytes of data.
64 bytes from slave2 (192.168.100.102): icmp_seq=1 ttl=64 time=1.90 ms
64 bytes from slave2 (192.168.100.102): icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from slave2 (192.168.100.102): icmp_seq=3 ttl=64 time=1.25 ms
64 bytes from slave2 (192.168.100.102): icmp_seq=4 ttl=64 time=1.31 ms
64 bytes from slave2 (192.168.100.102): icmp_seq=5 ttl=64 time=1.15 ms
```

3.设置免密登录

首先获取本节点的密钥

命令：ssh-keygen -t rsa

```
[root@master xurany]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:a+WJCx7ZYLOhKaMiYUAkDjkugZKPozaS0C/NyujMYJ0 root@master
The key's randomart image is:
+---[RSA 2048]----+
|=+               |
|@.               |
|+B               |
|*.o              |
|=o +  = S .      |
|=+o =+ B = .     |
|==+Eo = = o      |
|O.oo . + .       |
|=+     . .       |
+----[SHA256]-----+
[root@master xurany]#
```

分发密钥到master slave1 slave2:

命令：#ssh-copy-id master
#ssh-copy-id slave1
#ssh-copy-id slave2

这是master

```
[root@master xurany]# ssh-copy-id master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'master (192.168.100.100)' can't be established.
ECDSA key fingerprint is SHA256:iTHCYu9mPYkeewlhgSfu6Eca5ehQNZbbjKAHoJOD/7k.
ECDSA key fingerprint is MD5:d6:fe:d9:8f:4a:14:a6:f2:07:3b:e0:f0:b3:53:dc:66.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@master's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'master'"
and check to make sure that only the key(s) you wanted were added.

[root@master xurany]#
```

这是slave1

```
[root@master xurany]# ssh-copy-id slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave1 (192.168.100.101)' can't be established.
ECDSA key fingerprint is SHA256:Ennbu9UArTh7vywXzI5/rnNBrJJVd/Ft5SGGccTrXns.
ECDSA key fingerprint is MD5:7f:f8:c9:09:c9:9d:6d:37:de:f3:b8:e7:21:e4:ba:31.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave1'"
and check to make sure that only the key(s) you wanted were added.

[root@master xurany]#
```

这是slave2

```
[root@master xurany]# ssh-copy-id slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave2 (192.168.100.102)' can't be established.
ECDSA key fingerprint is SHA256:WNGbZ8+ukKoAw2DUATqtA+TK2xbAW/5etZpvCs+eZIQ.
ECDSA key fingerprint is MD5:6b:ca:36:e3:48:11:c8:2b:af:88:1c:33:89:63:b6:b7.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave2'"
and check to make sure that only the key(s) you wanted were added.

[root@master xurany]#
```

将/etc/hosts的配置分发到slave1和slave2，在slave1和slave2节点重复免密操作

命令：scp /etc/hosts slave1:/etc/

命令：scp /etc/hosts slave2:/etc/

```
[root@master xurany]# scp /etc/hosts slave1:/etc/
hosts                                         100%  227   134.7KB/s   00:00
[root@master xurany]# scp /etc/hosts slave2:/etc/
hosts                                         100%  227   120.5KB/s   00:00
[root@master xurany]#
```

这个是错误的

```
[root@master xurany]# scp /etc/hosts slave1:/etc/hosts
hosts                                         100%  227    97.4KB/s   00:00
```

Slave1上免密：

命令：ssh slave1

命令：ssh-keygen -t rsa

命令：ssh-copy-id master

命令：ssh-copy-id slave1

命令：ssh-copy-id slave2

```
[root@master xurany]# ssh slave1
Last login: Wed Nov  6 17:12:29 2024
[root@slave1 ~]# vi /etc/hosts
[root@slave1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7dHV9VCiUYDxeSpTj1Zxk7lCownqhyWx+xYYshRaom8 root@slave1
The key's randomart image is:
+---[RSA 2048]----+
|         .ooo=o*|
|   . o . ... * O+|
|  . + . + . O = +|
| . . o = o = O . |
|  . . + S = = o  |
|   E . = + =     |
|  .     o o      |
|         o       |
|        .        |
+----[SHA256]-----+
[root@slave1 ~]# ssh-copy-id master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'master (192.168.100.100)' can't be established.
ECDSA key fingerprint is SHA256:iTHCYu9mPYkeewlhgSfu6Eca5ehQNZbbjKAHoJOD/7k.
ECDSA key fingerprint is MD5:d6:fe:d9:8f:4a:14:a6:f2:07:3b:e0:f0:b3:53:dc:66.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@master's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'master'"
```

```
[root@slave1 ~]# ssh-copy-id slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave1 (192.168.100.101)' can't be established.
ECDSA key fingerprint is SHA256:Ennbu9UArTh7vywXzI5/rnNBrJJVd/Ft5SGGccTrXns.
ECDSA key fingerprint is MD5:7f:f8:c9:09:c9:9d:6d:37:de:f3:b8:e7:21:e4:ba:31.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave1'"
and check to make sure that only the key(s) you wanted were added.
```

```
[root@slave1 ~]# ssh-copy-id slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave2 (192.168.100.102)' can't be established.
ECDSA key fingerprint is SHA256:WNGbZ8+ukKoAw2DUATqtA+TK2xbAW/5etZpvCs+eZIQ.
ECDSA key fingerprint is MD5:6b:ca:36:e3:48:11:c8:2b:af:88:1c:33:89:63:b6:b7.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave2'"
and check to make sure that only the key(s) you wanted were added.

[root@slave1 ~]#
```

Slave2上免密：

命令：ssh slave2

命令：ssh-keygen -t rsa

命令：ssh-copy-id master

命令：ssh-copy-id slave1

命令：ssh-copy-id slave2

```
[root@slave1 ~]# ssh slave2
Last login: Wed Nov  6 17:14:54 2024
[root@slave2 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:f9UuZ1/bxQ9CxU4V/WJd3v+sKVXQuUM8T0VfAJBpq4o root@slave2
The key's randomart image is:
+---[RSA 2048]----+
|          .+..o=X|
|          +  ..*B|
|         . .  ==O|
|          .  +o+B|
|        S.  ..oo+|
|         .. . ..o.|
|     . .  . o.o.O|
|    E .    ... *X|
|             .oo+|
+----[SHA256]-----+
[root@slave2 ~]# ssh-copy-id master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'master (192.168.100.100)' can't be established.
ECDSA key fingerprint is SHA256:iTHCYu9mPYkeewlhgSfu6Eca5ehQNZbbjKAHoJOD/7k.
ECDSA key fingerprint is MD5:d6:fe:d9:8f:4a:14:a6:f2:07:3b:e0:f0:b3:53:dc:66.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@master's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'master'"
and check to make sure that only the key(s) you wanted were added.

[root@slave2 ~]#
```

```
[root@slave2 ~]# ssh-copy-id slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave1 (192.168.100.101)' can't be established.
ECDSA key fingerprint is SHA256:Ennbu9UArTh7vywXzI5/rnNBrJJVd/Ft5SGGccTrXns.
ECDSA key fingerprint is MD5:7f:f8:c9:09:c9:9d:6d:37:de:f3:b8:e7:21:e4:ba:31.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave1'"
and check to make sure that only the key(s) you wanted were added.

[root@slave2 ~]# ssh-copy-id slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave2 (192.168.100.102)' can't be established.
ECDSA key fingerprint is SHA256:WNGbZ8+ukKoAw2DUATqtA+TK2xbAW/5etZpvCs+eZIQ.
ECDSA key fingerprint is MD5:6b:ca:36:e3:48:11:c8:2b:af:88:1c:33:89:63:b6:b7.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave2'"
and check to make sure that only the key(s) you wanted were added.

[root@slave2 ~]#
```

4.三个节点都要查看是否有防火墙开启，如果有就关闭且禁用防火墙

查看是否有防火墙开启

命令：systemctl status firewalld

```
[root@master xurany]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since 三 2024-11-06 15:33:55 CST; 3h 0min ago
     Docs: man:firewalld(1)
 Main PID: 964 (firewalld)
    Tasks: 2
   CGroup: /system.slice/firewalld.service
           └─964 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

11月 06 15:33:55 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
11月 06 15:33:55 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
11月 06 15:33:55 localhost.localdomain firewalld[964]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in a future release. Pleas...bling it now.
Hint: Some lines were ellipsized, use -l to show in full.
[root@master xurany]#
```

关闭防火墙

命令：systemctl stop firewalld

关闭后在查看会显示这个

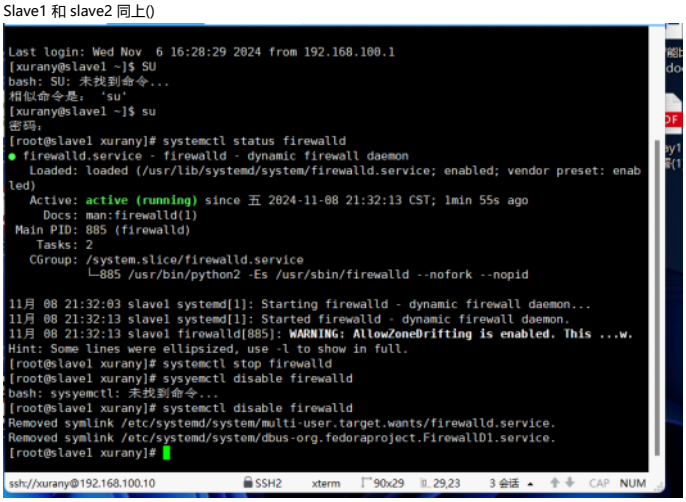
```
[root@master xurany]# systemctl stop firewalld
[root@master xurany]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since 三 2024-11-06 18:37:46 CST; 15s ago
     Docs: man:firewalld(1)
  Process: 964 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
 Main PID: 964 (code=exited, status=0/SUCCESS)

11月 06 15:33:55 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
11月 06 15:33:55 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
11月 06 15:33:55 localhost.localdomain firewalld[964]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in a future release. Pleas...bling it now.
11月 06 18:37:45 master systemd[1]: Stopping firewalld - dynamic firewall daemon...
11月 06 18:37:46 master systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@master xurany]#
```

禁用防火墙

命令：systemctl disable firewalld

```
[root@master xurany]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@master xurany]#
```

Slave1 和 slave2 同上()




```
[xurany@slave2 ~]$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since 五 2024-11-08 21:32:13 CST; 4min 39s ago
     Docs: man:firewalld(1)
 Main PID: 899 (firewalld)
    Tasks: 2
   CGroup: /system.slice/firewalld.service
           └─899 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid
[xurany@slave2 ~]$ system stop firewalld
bash: system: 未找到命令...
[xurany@slave2 ~]$ systemctl stop firewalld
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to manage system services or units.
Authenticating as: root
Password:
==== AUTHENTICATION COMPLETE ===
[xurany@slave2 ~]$ su
密码：
[root@slave2 xurany]# systemctl stop firewalld
[root@slave2 xurany]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@slave2 xurany]#
```

2.在集群上安装jdk和hadoop

在三个节点创建/opt/module目录

命令：mkdir /opt/module
命令：scp -r /opt/module slave1:/opt/
命令：scp -r /opt/module slave2:/opt/
命令：ssh slave1
命令：ls /opt
命令：exit

```
Connecting to 192.168.100.100:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Last login: Wed Nov 6 16:27:09 2024 from 192.168.100.1
[xurany@master ~]$ su
密码：
[root@master xurany]# mkdir /opt/module/
[root@master xurany]# scp /opt/module/ slave1:/opt/
/opt/module: not a regular file
[root@master xurany]# scp -r /opt/module/ slave1:/opt/
[root@master xurany]# scp -r /opt/module/ slave2:/opt/
[root@master xurany]# ssh slave1
Last login: Fri Nov 8 21:33:49 2024
[root@slave1 ~]# ls /opt/
module rh
[root@slave1 ~]# exit
登出
Connection to slave1 closed.
[root@master xurany]# ssh slave2
Last login: Fri Nov 8 21:37:44 2024
[root@slave2 ~]# ls /opt/
module rh
[root@slave2 ~]# exit
登出
Connection to slave2 closed.
[root@master xurany]#
```

在master节点安装jdk：
命令：tar -zxvf /opt/softwares/jdk???.tar.gz -C /opt/module

```
[root@master softwares]# cd /home/xurany/
[root@master xurany]# tar -zxvf /opt/softwares/jdk-8u212-linux-x64.tar.gz -C /opt/module/
jdk1.8.0_212/
jdk1.8.0_212/README.html
jdk1.8.0_212/LICENSE
jdk1.8.0_212/include/
jdk1.8.0_212/include/jawt.h
jdk1.8.0_212/include/linux/
jdk1.8.0_212/include/linux/jawt_md.h
jdk1.8.0_212/include/linux/jni_md.h
jdk1.8.0_212/include/classfile_constants.h
jdk1.8.0_212/include/jvmticmlr.h
jdk1.8.0_212/include/jni.h
jdk1.8.0_212/include/jdwpTransport.h
jdk1.8.0_212/include/jvmti.h
jdk1.8.0_212/THIRDPARTYLICENSEREADME-JAVAFX.txt
jdk1.8.0_212/lib/
```

查看/opt/目录确定jdk被成功解压

#命令：ls /opt/module/

```
[root@master xurany]# ls /opt/module/
jdk1.8.0_212
```

配置jdk的环境变量：

命令：vim /etc/profile

```
[root@master jdk]# vim /etc/profile
```

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk
export PATH="$JAVA_HOME/bin:$PATH"
                                                              67,1          底端
```

配置内容：

#JAVA_HOME
export JAVA_HOME=/opt/module/jdk
export PATH="$JAVA_HOME/bin:$PATH"

source使配置文件生效，并且查看jdk版本是否正确

命令：source /etc/profile

```
[root@master jdk]# source /etc/profile
[root@master jdk]#
```

命令：java -version

```
[root@master jdk]# java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
[root@master jdk]#
```

将jdk安装到slave1和slave2:

拷贝/etc/profile

命令：scp /etc/profile slave1:/etc

命令：scp /etc/profile slave2:/etc

```
[root@master jdk]# cd /home/xurany/
[root@master xurany]# cd ~
[root@master ~]# scp /etc/profile slave1:/etc/
profile                                       100% 1898     2.0MB/s   00:00
[root@master ~]# scp /etc/profile slave2:/etc/
profile                                       100% 1898     1.9MB/s   00:00
[root@master ~]#
```

拷贝jdk:

命令：scp -r /opt/module/jdk/ slave1:/opt/module/

```
[root@master ~]# scp -r /opt/module/jdk/ slave1:/opt/module/
README.html                                   100%  159   230.0KB/s   00:00
LICENSE                                       100%   44    59.9KB/s   00:00
jawt.h                                        100% 8690     9.5MB/s   00:00
jawt_md.h                                     100%  995     1.3MB/s   00:00
jni_md.h                                      100%  824     1.5MB/s   00:00
classfile_constants.h                         100%   20KB  11.3MB/s   00:00
jvmticmlr.h                                   100% 3774     4.7MB/s   00:00
jni.h                                         100%   72KB  32.8MB/s   00:00
jdwpTransport.h                               100% 6407     6.8MB/s   00:00
jvmti.h                                       100%   76KB  42.7MB/s   00:00
THIRDPARTYLICENSEREADME-JAVAFX.txt            100%  110KB  54.3MB/s   00:00
ct.sym                                        100%   17MB  91.9MB/s   00:00
orb.idl                                       100%  640     1.0MB/s   00:00
artifacts.xml                                 100%   89KB  54.4MB/s   00:00
README.TXT                                    100%  168   309.7KB/s   00:00
org.eclipse.osgi.compatibility.state.nl_zh_4.4.0.v20140 100% 3156     4.4MB/s   00:00
org.eclipse.osgi.compatibility.state.nl_ja_4.4.0.v20140 100% 3313     4.3MB/s   00:00
org.eclipse.equinox.p2.ui.sdk.scheduler_1.2.0.v20140422 100%  112KB  45.0MB/s   00:00
org.eclipse.swt.nl_ja_4.4.0.v20140623020002.jar       100% 3559     4.1MB/s   00:00
org.eclipse.equinox.ds_1.4.200.v20131126-2331.jar     100%  190KB  30.1MB/s   00:00
org.eclipse.ecf.provider.filetransfer.httpclient4.ssl_1 100%   13KB  10.2MB/s   00:00
org.eclipse.e4.ui.css.core_0.10.100.v20140424-2042.jar  100%  202KB  31.7MB/s   00:00
org.apache.jasper.glassfish_2.2.2.v201205150955.jar   100% 2328KB  79.9MB/s   00:00
```

命令：scp -r /opt/module/jdk/ slave2:/opt/module/

```
[root@master ~]# scp -r /opt/module/jdk/ slave2:/opt/module/
README.html                                          100%  159   163.1KB/s   00:00
LICENSE                                              100%   44    71.5KB/s   00:00
jawt.h                                               100% 8690    10.2MB/s   00:00
jawt_md.h                                            100%  995     1.5MB/s   00:00
jni_md.h                                             100%  824     1.6MB/s   00:00
classfile_constants.h                                100%   20KB  17.2MB/s   00:00
jvmticmlr.h                                          100% 3774     5.3MB/s   00:00
jni.h                                                100%   72KB  42.7MB/s   00:00
jdwpTransport.h                                      100% 6407     8.6MB/s   00:00
jvmti.h                                              100%   76KB  43.7MB/s   00:00
THIRDPARTYLICENSEREADME-JAVAFX.txt                   100%  110KB  33.9MB/s   00:00
ct.sym                                               100%   17MB  98.1MB/s   00:00
orb.idl                                              100%  640   276.1KB/s   00:00
artifacts.xml                                        100%   89KB  43.0MB/s   00:00
README.TXT                                           100%  168   140.5KB/s   00:00
org.eclipse.osgi.compatibility.state.nl_zh_4.4.0.v20140 100% 3156     3.7MB/s   00:00
org.eclipse.osgi.compatibility.state.nl_ja_4.4.0.v20140 100% 3313     5.1MB/s   00:00
org.eclipse.equinox.p2.ui.sdk.scheduler_1.2.0.v20140422 100%  112KB  53.5MB/s   00:00
org.eclipse.swt.nl_ja_4.4.0.v20140623020002.jar      100% 3559     3.7MB/s   00:00
org.eclipse.equinox.ds_1.4.200.v20131126-2331.jar    100%  190KB  50.5MB/s   00:00
org.eclipse.ecf.provider.filetransfer.httpclient4.ssl_1 100%   13KB  13.4MB/s   00:00
org.eclipse.e4.ui.css.core_0.10.100.v20140424-2042.jar 100%  202KB  66.1MB/s   00:00
org.apache.jasper.glassfish_2.2.2.v201205150955.jar  100% 2328KB  88.4MB/s   00:00
org.eclipse.equinox.jsp.jasper_1.0.400.v20130327-1442.j 100%   27KB   7.6MB/s   00:00
org.eclipse.swt.gtk.linux.x86_64_3.103.1.v20140903-1947 100% 2536KB 107.9MB/s   00:00
org.eclipse.ui.themes.nl_ja_4.4.0.v20140623020002.jar 100% 1516     2.2MB/s   00:00
org.eclipse.jface.nl_ja_4.4.0.v20140623020002.jar    100% 5985     8.7MB/s   00:00
```

在slave1和slave2上source /etc/profile 和 java -version ,检查是否安装jdk成功

命令：source /etc/profile

命令：java -version

Slave1:

```
[root@slave1 xurany]# source /etc/profile
[root@slave1 xurany]# java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
[root@slave1 xurany]#
```

Slave2:

```
[root@slave2 xurany]# source /etc/profile
[root@slave2 xurany]# java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
[root@slave2 xurany]#
```

3.安装hadoop分布式

1.解压hadoop安装包（此路径在softwares下）

命令：tar -zxvf hadoop3.13.tar.gz -C /opt/module

```
hadoop-3.1.3.tar.gz  jdk-8u212-linux-x64.tar.gz  L???s???CAR:k??_?
[root@master softwares]# tar -zxvf hadoop-3.1.3.tar.gz -C /opt/module/
```

改个名，改成hadoop

命令：mv hadoop-3.1.3 hadoop

```
[root@master module]# mv hadoop-3.1.3 hadoop
[root@master module]# ls
hadoop  jdk
```

2.配置hadoop的环境变量并生效（在hadoop路径下）

命令：vim /etc/profile

```
# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk
export PATH="$JAVA_HOME/bin:$PATH"
#HADOOP_HOME
export HADOOP_HOME=/opt/module/hadoop
export PATH="$HADOOP_HOME/bin:$PATH"
"/etc/profile" 82L, 1986C                                     82,1          底端
```

配置文件：

#HADOOP_HOME

export HADOOP_HOME=/opt/module/hadoop

export PATH="$HADOOP_HOME/bin:$PATH"

使环境变量生效：

命令：source /etc/profile

验证hadoop

命令：hadoop version

```
[root@master hadoop]# vim /etc/profile
[root@master hadoop]# source /etc/profile
[root@master hadoop]# hadoop version
Hadoop 3.1.3
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r ba631c436b806728f
8ec2f54ab1e289526c90579
Compiled by ztang on 2019-09-12T02:47Z
Compiled with protoc 2.5.0
From source with checksum ec785077c385118ac91aadde5ec9799
This command was run using /opt/module/hadoop/share/hadoop/common/hadoop-common-3.1.3.jar
[root@master hadoop]#
```

3.进入到hadoop的配置文件路径，配置Hadoop的配置文件

命令：cd /opt/module/hadoop/etc/hadoop

```
[root@master hadoop]# cd ../
[root@master module]# cd /opt/module/hadoop/etc/hadoop/
[root@master hadoop]# ls
capacity-scheduler.xml            kms-log4j.properties
configuration.xsl                 kms-site.xml
container-executor.cfg            log4j.properties
core-site.xml                     mapred-env.cmd
hadoop-env.cmd                    mapred-env.sh
hadoop-env.sh                     mapred-queues.xml.template
hadoop-metrics2.properties        mapred-site.xml
hadoop-policy.xml                 shellprofile.d
hadoop-user-functions.sh.example  ssl-client.xml.example
hdfs-site.xml                     ssl-server.xml.example
httpfs-env.sh                     user_ec_policies.xml.template
httpfs-log4j.properties           workers
httpfs-signature.secret           yarn-env.cmd
httpfs-site.xml                   yarn-env.sh
kms-acls.xml                      yarnservice-log4j.properties
kms-env.sh                        yarn-site.xml
[root@master hadoop]# vim hadoop-env.sh
```

1) 配置Hadoop-env.sh

命令：vim hadoop-env.sh

```
[root@master hadoop]# vim hadoop-env.sh
```

```
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HDFS_DFSROUTER_OPTS=""
###

###
# Advanced Users Only!
###

#
# When building Hadoop, one can add the class paths to the commands
# via this special env var:
# export HADOOP_ENABLE_BUILD_PATHS="true"

#
# To prevent accidents, shell commands be (superficially) locked
# to only allow certain users to execute certain subcommands.
# It uses the format of (command)_(subcommand)_USER.
#
# For example, to limit who can execute the namenode command,
# export HDFS_NAMENODE_USER=hdfs
export JAVA_HOME=/opt/module/jdk
export HDFS_NAMENODE_USER=root
export HDFS_DATANODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export YARN_RESOURCEMANAGER_USER=root
export YARN_NODEMANAGER_USER=root
"hadoop-env.sh" 415L, 16110C                                  415,1         底端
```

配置内容

export JAVA_HOME=/opt/module/jdk
export HDFS_NAMENODE_USER=root
export HDFS_DATANODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
export YARN_RESOURCEMANAGER_USER=root
export YARN_NODEMANAGER_USER=root

2)配置core-site.xml

命令：vim core-site.xml

```
[root@master hadoop]# vim core-site.xml
```

```
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<!--指定NameNode的地址-->
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:8020</value>
</property>
<!--指定hadoop数据的存储目录-->
<property>
        <name>hadoop.tmp.dir</name>
        <value>/opt/module/hadoop/data</value>
</property>
<!-- 配置HDFS网页登录使用的静态用户root-->
<property>
        <name>hadoop.http.staticuser.user</name>
        <value>root</value>
</property>
<!-- 配置root(superUser)允许通过代理访问主机节点-->
<property>
        <name>hadoop.proxyuser.root.hosts</name>
        <value>*</value>
</property>
<!--配置root(superUser)允许代理用户所属组-->
<property>
        <name>hadoop.proxyuser.root.groups</name>
        <value>*</value>
</property>
<!--配置root(superUser)允许通过代理的用户-->
<property>
        <name>hadoop.proxyuser.root.users</name>
        <value>*</value>
</property>
</configuration>
"core-site.xml" 50L, 1594C
```

配置文件

```
<configuration>
<!--指定NameNode的地址-->
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:8020</value>
    </property>
<!--指定hadoop数据的存储目录-->
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/opt/module/hadoop/data</value>
```

```
    </property>
<!--配置HDFS网页登陆使用的静态用户 root-->
    <property>
        <name>hadoop.http.staticuser.user</name>
        <value>root</value>
    </property>
<!--配置root（superUser)允许通过代理访问主机节点-->
    <property>
        <name>hadoop.proxyuser.root.hosts</name>
        <value>*</value>
    </property>
<!--配置root（superUser)允许代理用户所属组-->
    <property>
        <name>hadoop.proxyuser.root.groups</name>
        <value>*</value>
    </property>
<!--配置root(superUser)允许通过代理的用户-->
    <property>
        <name>hadoop.proxyuser.root.users</name>
        <value>*</value>
    </property>
</configuration>
```

3.)配置hdfs-site.xml

命令：vim hdfs-site.xml

```
[root@master hadoop]# vim hdfs-site.xml
```

```
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<!--nn web端访问地址-->
        <property>
                <name>dfs.nnamenide.http-address</name>
                <value>master:9870</value>
        </property>
<!--2nn web端访问地址-->
        <property>
                <name>dfs.namenode.secondary.http-address</name>
                <value>slave2:9868</value>
        </property>
<!--指定集群环境HDFS副本的数量-->
        <property>
                <name>dfs.replication</name>
                <value>>1</value>
        </property>

</configuration>
~
~
~
~
~
~
                                                                34,2-9        底端
```

配置文件：

```
<configuration>
<!--nn web端访问地址-->
    <property>
        <name>dfs.namenode.http-address</name>
        <value>master:9870</value>
    </property>
<!--2nn web端访问地址-->
    <property>
        <name>dfs.namenode.secondary.http-address</name>
        <value>slave2:9868</value>
    </property>
<!--指定集群环境HDFS副本的数量-->
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
</configuration>
```

4)配置mapred-site.xml

命令：vim mapred-site.xml

```
[root@master hadoop]# vim mapred-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<!--指定MapReduce程序在Yarn上运行-->
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
<!--配置历史服务器端地址-->
        <property>
                <name>mapreduce.jobhistory.address</name>
                <value>master:10020</value>
        </property>
<!--配置历史服务器Web端地址-->
        <property>
                <name>mapreduce.jobhistory.wabapp.address</name>
                <value>master:19888</value>
        </property>

</configuration>
~
~
~
"mapred-site.xml" 36L, 1174C                                          36,9          全部
```

配置文件：

```
<configuration>
<!--指定MapReduce程序在Yarn上运行-->
      <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
      </property>
<!--配置历史服务器地址-->
      <property>
            <name>mapreduce.jobhistory.address</name>
            <value>master:10020</value>
      </property>
<!--配置历史服务器Web端地址-->
      <property>
            <name>mapreduce.jobhistory.webapp.address</name>
            <value>master:19888</value>
      </property>
</configuration>
```

5)配置yarn-site.xml

命令：vim yarn-site.xml

```
<configuration>

<!-- Site specific YARN configuration properties -->
<!--指定MR走shuffle-->
        <property>
                        <name>yarn.nodemanager.aux-services</name>
                        <value>mapreduce_shuffle</value>
        </property>
<!--指定ResourceManager的地址-->
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>slave1</value>
        </property>
<!--环境变量的继承-->
        <property>
                <name>yarn.nodemanager.env-whitelist</name>
                <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,C LASSPATH_PREPEND
_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
        </property>
<!--yarn单个容器允许分配的最大内存大小-->
        <property>
                <name>yarn.scheduler.minimum-allocation-mb</name>
                <value>512</value>
        </property>
        <property>
                <name>yarn.scheduler.maximux-allocation-mb</name>
                <value>4096</value>
        </property>
<!--关闭yarn对物理内存和虚拟内存的限制检查-->
        <property>
                <name>yarn.nodemanager.pmem-check-enabled</name>
                <value>false</value>
        </property>
        <property>
                <name>yarn.nodemanager.vmem-check-enabled</name>
                <value>false</value>
        </property>

</configuration>
"yarn-site.xml" 52L, 1742C                                            52,1          底端
```

配置文件

```
<!--指定MR走shuffle-->
      <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
      </property>
<!--指定ResourceManager的地址-->
      <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>slave1</value>
      </property>
<!--环境变量的继承-->
      <property>
            <name>yarn.nodemanager.env-whitelist</name>
            <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,C
            LASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
```

```
    </property>
<!--yarn单个容器允许分配的最大内存大小-->
    <property>
        <name>yarn.scheduler.minimum-allocation-mb</name>
        <value>512</value>
    </property>
    <property>
        <name>yarn.scheduler.maximux-allocation-mb</name>
        <value>4096</value>
    </property>
<!--关闭yarn对物理内存和虚拟内存的限制检查-->
    <property>
        <name>yarn.nodemanager.pmem-check-enabled</name>
        <value>false</value>
    </property>
    <property>
        <name>yarn.nodemanager.vmem-check-enabled</name>
        <value>false</value>
    </property>
```

6)配置workers文件

命令：vim workers

```
[root@master hadoop]# vim workers
```

```
localhost
master
slave1
slave2
~
~
```

配置内容

master
slave1
slave2:

7)将Hadoop节点拷贝到slave1和slave2

命令：scp -r /opt/module/hadoop/ slave1:/opt/module/

命令：scp -r /opt/module/hadoop/ slave2:/opt/module/

命令：scp /etc/profile slave1:/etc/

命令：scp /etc/profile slave2:/etc/

弄完记得三台机子刷新环境变量

命令：source /etc/profile

格式化NAMENODE 启动hadoop集群

命令：hdfs namenode -format

```
[root@master hadoop]# hdfs namenode -format
2024-11-10 01:03:31,159 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/192.168.188.160
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.1.3
STARTUP_MSG:   classpath = /opt/module/hadoop/etc/hadoop:/opt/module/hadoop/share/hadoop/common/lib/acc
essors-smart-1.2.jar:/opt/module/hadoop/share/hadoop/common/lib/animal-sniffer-annotations-1.17.jar:/op
t/module/hadoop/share/hadoop/common/lib/asm-5.0.4.jar:/opt/module/hadoop/share/hadoop/common/lib/audien
ce-annotations-0.5.0.jar:/opt/module/hadoop/share/hadoop/common/lib/avro-1.7.7.jar:/opt/module/hadoop/s
hare/hadoop/common/lib/checker-qual-2.5.2.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-beanut
ils-1.9.3.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/opt/module/hadoop/share/h
adoop/common/lib/commons-codec-1.11.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-collections-
3.2.2.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-compress-1.18.jar:/opt/module/hadoop/share
/hadoop/common/lib/commons-configuration2-2.1.1.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-
io-2.5.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-lang-2.6.jar:/opt/module/hadoop/share/had
oop/common/lib/commons-lang3-3.4.jar:/opt/module/hadoop/share/hadoop/common/lib/commons-logging-1.1.3.j
ar:/opt/module/hadoop/share/hadoop/common/lib/commons-math3-3.1.1.jar:/opt/module/hadoop/share/hadoop/c
ommon/lib/commons-net-3.6.jar:/opt/module/hadoop/share/hadoop/common/lib/curator-client-2.13.0.jar:/opt
/module/hadoop/share/hadoop/common/lib/curator-framework-2.13.0.jar:/opt/module/hadoop/share/hadoop/com
mon/lib/curator-recipes-2.13.0.jar:/opt/module/hadoop/share/hadoop/common/lib/error_prone_annotations-2
.2.0.jar:/opt/module/hadoop/share/hadoop/common/lib/failureaccess-1.0.jar:/opt/module/hadoop/share/hado
op/common/lib/gson-2.2.4.jar:/opt/module/hadoop/share/hadoop/common/lib/guava-27.0-jre.jar:/opt/module/
hadoop/share/hadoop/common/lib/hadoop-annotations-3.1.3.jar:/opt/module/hadoop/share/hadoop/common/lib/
hadoop-auth-3.1.3.jar:/opt/module/hadoop/share/hadoop/common/lib/htrace-core4-4.1.0-incubating.jar:/opt
/module/hadoop/share/hadoop/common/lib/httpclient-4.5.2.jar:/opt/module/hadoop/share/hadoop/common/lib/
httpcore-4.4.4.jar:/opt/module/hadoop/share/hadoop/common/lib/j2objc-annotations-1.1.jar:/opt/module/ha
doop/share/hadoop/common/lib/jackson-annotations-2.7.8.jar:/opt/module/hadoop/share/hadoop/common/lib/j
ackson-core-2.7.8.jar:/opt/module/hadoop/share/hadoop/common/lib/jackson-core-asl-1.9.13.jar:/opt/modul
```

4.启动hadoop集群

master节点

命令：cd /opt/module/hadoop

命令：./sbin/start-dfs.sh

命令：jps

```
[root@master hadoop]# jps
14177 NameNode
3413 NodeManager
14437 DataNode
15371 Jps
[root@master hadoop]#
```

slave1节点：

命令：./sbin/start-yarn.sh

命令：jps

```
[root@slave1 hadoop]# ./sbin/start-yarn.sh
Starting resourcemanager
上一次登录：六 11月  9 19:07:37 CST 2024pts/0 上
resourcemanager is running as process 13374.  Stop it first.
Starting nodemanagers
上一次登录：日 11月 10 01:08:03 CST 2024pts/0 上
localhost: nodemanager is running as process 18655.  Stop it first.
slave1: nodemanager is running as process 18655.  Stop it first.
slave2: nodemanager is running as process 18824.  Stop it first.
[root@slave1 hadoop]# jps
18514 DataNode
19202 Jps
13374 ResourceManager
18655 NodeManager
[root@slave1 hadoop]#
```

要停止的话：

在master

命令：./sbin/stop-all.sh
命令：./sbin/stop-yarn.sh