

第 2 章 Hadoop 集群环境搭建

★本章导读★

从本章就开始了大数据的正式旅程。本章节会首先从 Hadoop 开始，介绍 Hadoop 大数据集群的结构，并手把手从零开始介绍虚拟机、Linux 系统创建以及 Hadoop 大数据集群的搭建。通过本章的学习，读者能掌握 Hadoop 集群的搭建部署，能快速揭示 Hadoop 的真面目。

★知识要点★

通过本章内容的学习，读者将掌握以下知识：

- 掌握虚拟机的安装及配置
- 掌握 Linux 环境的基本使用
- 掌握 Hadoop 完全分布式环境的搭建
- 了解 Hadoop UI 的使用

2.1 Hadoop 集群环境介绍

为了更深入的了解大数据环境，需要读者搭建 Hadoop 完全分布式集群环境。Hadoop 集群环境可以分为单机版、伪分布式版和完全分布式版环境。单机版环境是指在一台单机上运行 Hadoop；伪分布式环境可以看作是在一台机器上模拟分布式环境；完全分布式环境则是在多台机器上组成的 Hadoop 集群环境。

2.1.1 Hadoop 组件选择

本章节将详细地介绍 Hadoop 完全分布式环境的搭建，开启属于自己的大数据之门。对于个人计算机环境，建议计算机硬件最低配置有如下要求：

1. 硬盘容量大于等于 100GB；
2. 运行内存容量大于等于 8G；
3. 计算机 CPU 为 Intel i3 以上的处理器。

下面对所搭建的 Hadoop 完全分布式环境所涉及的相关软件版本信息进行介绍，主要的软件和版本如表 2-1 所示。

表 2-1 Hadoop 环境相关软件及版本

| 软件名称 | 版本 | 安装包名称 |
|--------------------|------------|------------------------------|
| Linux | CentOS 7.9 | CentOS-7-x86_64-DVD-2009.iso |
| VMware WorkStation | 15 | VMware Workstation V15. |
| JDK | 1.8.0 | jdk-8u172-linux-x64.tar.gz |
| Hadoop | 2.6.5 | hadoop-2.6.5.tar.gz |

本书采用 VMware 虚拟机软件作为大数据集群环境搭建的基石，通过虚拟机安装 Linux 环境来模拟部署服务器环境。本书所选择相关软件详情如下所示。

1. 本书选择了 CentOS 7.9 的 Linux 操作系统作为集群的服务器系统, 因为 CentOS 系统和 Ubuntu 系统相比, CentOS 系统更为稳定, 所以更适合作为服务器操作系统, 且现在工业界中多使用较为稳定的 CentOS 7.x 版本。
2. 在软件选择方面, 集群环境中的 Java 软件选择了使用较为广泛且稳定的 JDK 1.8.0
3. Hadoop 完全分布式环境中最关键的 Hadoop 软件, 本书选择了 Hadoop 2.6.5, 同样是因为在现在大数据生产环境中, Hadoop 2.6~Hadoop 2.8 是企业优先选择的稳定版本。

2.1.2 Hadoop 集群节点配置

由第一章介绍可知, Hadoop 完全分布式环境采用的是 Master/Slave 架构, 本章节所搭建的 Hadoop 完全分布式集群拓扑结构如图 2-1 所示。

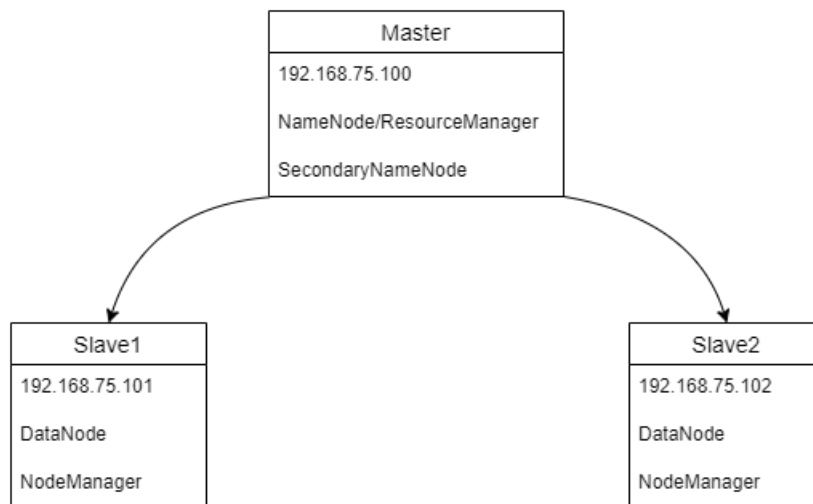


图 2-1 Hadoop 完全分布式集群拓扑结构图

温馨提示：

拓扑图中各个节点标识的 IP 地址 为了方便开发使用 在后续介绍中

2.2 VMmare 虚拟机安装与配置

VMware WorkStation 是 VMware 公司推出的一款功能非常全面的虚拟机软件, 它可以满足用户在现有主机操作系统中灵活地搭建、运行其他操作系统, 并且不干扰主机操作系统的正常运行。本书选择的了 VMware WorkStation 15.5 Pro 版本的虚拟机软件, VMware 软件的安装也比较简单, 下面进行详细介绍。

2.2.1 虚拟机软件安装

首先, 我们需要下载 VMware 安装程序。

进入 VMware 官网 “<https://www.vmware.com/>” 后, 可以选择并找到 VMware Workstation 产品进行下载, 也可以通过本书提供的软件直接下载使用。安装 VMware

Workstation 的过程比较简单，只需要在安装过程中选择合适的安装目录，然后按着软件提示依次单击【下一步】按钮进行安装，最后输入产品序列号，即可完成 VMware 的安装。安装完成后打开虚拟机软件如图 2-2 所示。

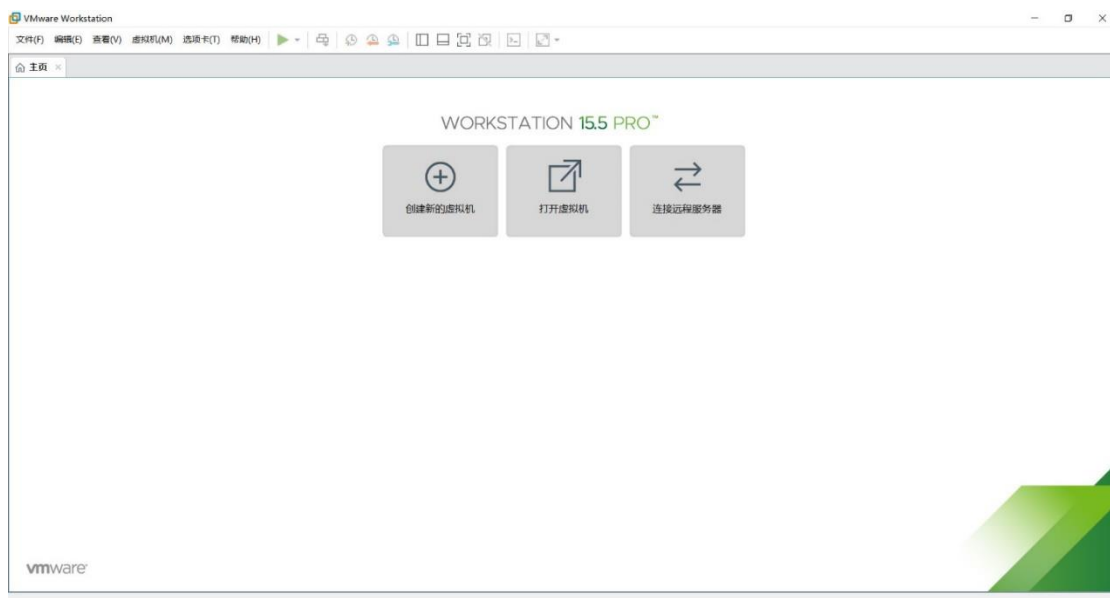


图 2-2 打开虚拟机软件图

2.2.2 创建 Linux 虚拟机

接下来，在成功安装 VMware 虚拟机软件之后，我们就可以进行 CentOS 7.9 的 Linux 操作系统的安装了。安装步骤如下：

步骤 01：打开 VMware 软件，在主页单击选择【创建新的虚拟机】选项，如图 2-3 所示，由此进入到虚拟机创建的步骤引导环节。

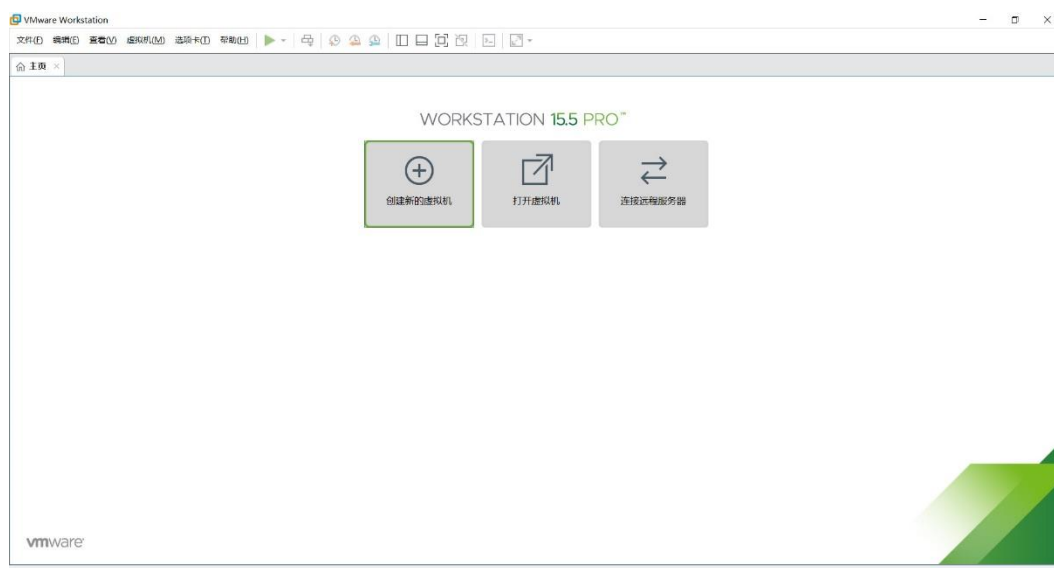


图 2-3 选择【创建新的虚拟机】选项图



图 2-4 选择配置模式

步骤 02: 在进入“新建虚拟机向导”页面之后，选择【自定义(高级)(C)】选项卡，接着单击【下一步】按钮，如图 2-4 所示。

步骤 03: 在该步骤中，在硬件兼容性下拉菜单中选择【Workstation 15.x】，然后单击【下一步】按钮，如图 2-5 所示。

步骤 04: 选择客户机操作系统，选中【安装程序光盘映像文件(iso)(M)】选项卡，并单击后方【浏览】按钮，选择自己下载的 CentOS 操作系统镜像所在位置，CentOS 镜像可以通过 <https://www.centos.org/download/> 官网进行下载，同样也可以直接选择通过本书附带的镜像文件下载。选择好镜像文件之后，继续回到安装客户机操作系统页面，并单击【下一步】，如图 2-6 所示。

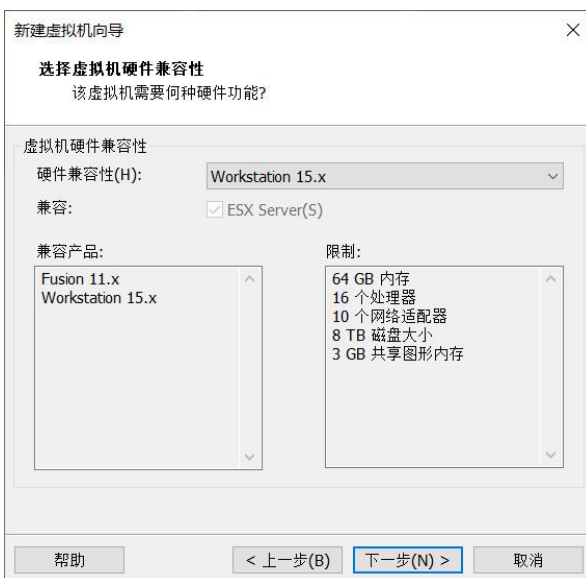


图 2-5 选择硬件兼容性

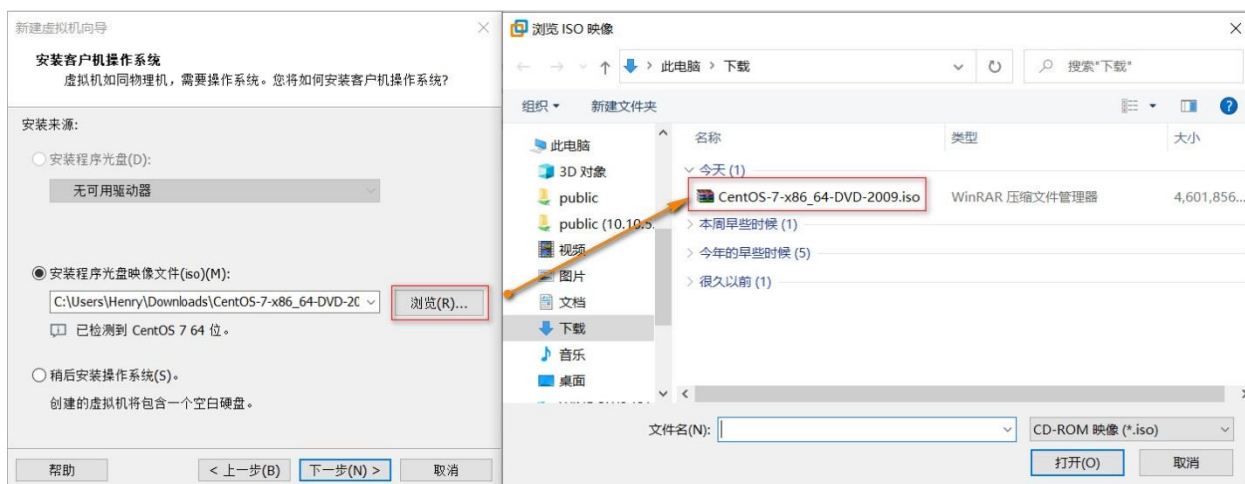


图 2-6 安装客户机操作系统

步骤 05: 为此虚拟机配置内存大小，通过滑动标尺选择到【2 GB】，或者在右边输入框中输入【2048】，该配置指定了虚拟机运行内存情况，如图 2-7 所示。在后续使用中可以进行更改，如果计算机内存较小，也可以选择系统推荐的【1 GB】内存，选择完成之后单击【下一步】。

步骤 06: 网络类型配置，选择默认的【使用网络地址转换(NAT)(E)】即可，如图 2-8 所示，然后继续单击【下一步】。

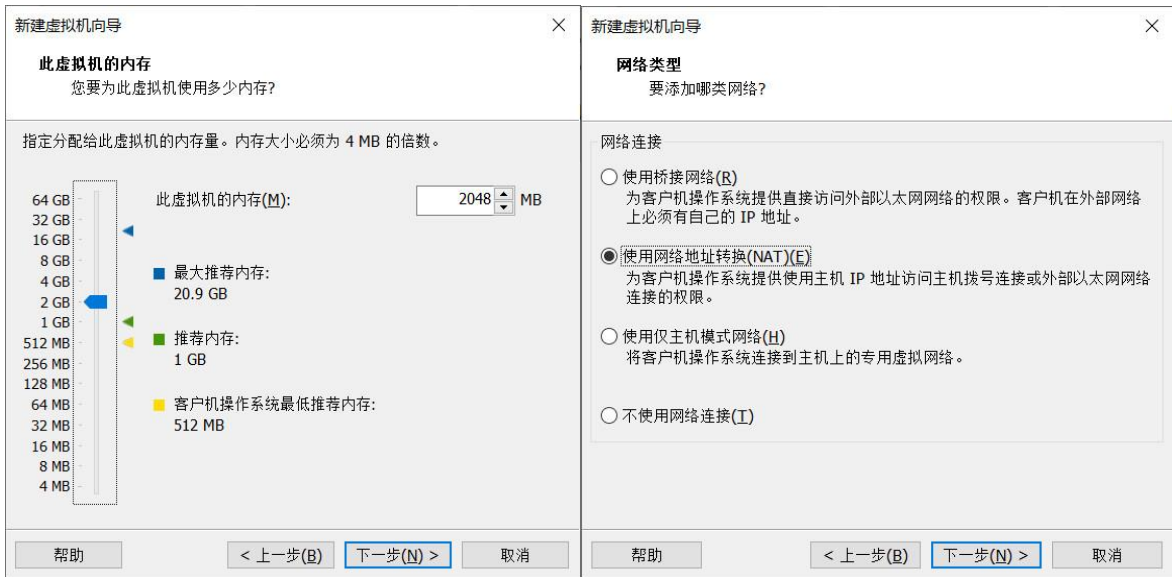


图 2-7 内存设置

图 2-8 网络类型

步骤 07：选择 I/O 控制器类型，同样，选择默认的【LSI Logic(L)(推荐)】选项即可，选择完成之后单击【下一步】，如图 2-9 所示。

步骤 08：选择磁盘类型，该步是为确认虚拟机选择创建哪一类型的磁盘，这里选择【SCSI(S)(推荐)】选项，如图 2-10 所示，然后单击【下一步】。



图 2-9 I/O 控制器类型

图 2-10 磁盘类型

步骤 09：选择磁盘，由于我们这里是创建全新的虚拟机，所以选择【创建新虚拟机磁盘(V)】选项，选择完成之后单击【下一步】，如图 2-11 所示。

步骤 10：指定磁盘容量，这一步是为虚拟机设定磁盘的大小，这里推荐设置【50】GB，并在下方选项卡中选择【将虚拟磁盘拆分成多个文件(M)】以方便我们后期灵活的移动，这里我们没有选择【立即分配所有磁盘空间(A)】，所以不必担心虚拟机会一下占用计算机 50 GB 的硬盘空间，磁盘容量会随着虚拟机内容的增加而增加，虚拟机磁盘容量的上限即为我们设置的 50 GB，如图 2-12 所示，然后单击【下一步】。

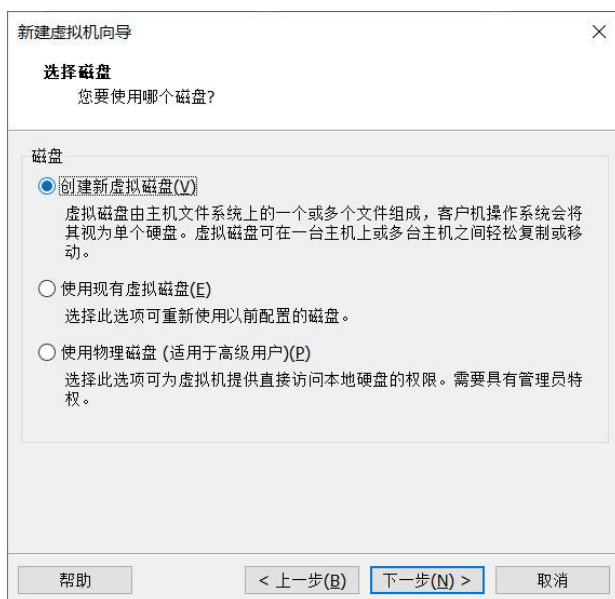


图 2-11 创建虚拟机磁盘

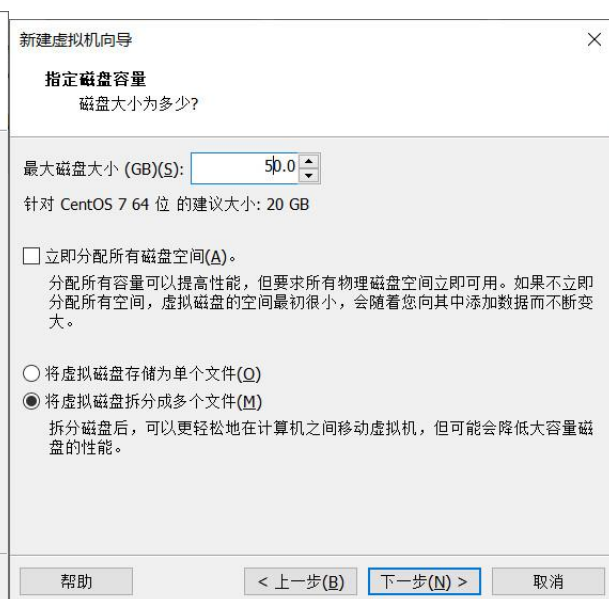


图 2-12 指定磁盘容量

步骤 11: 指定磁盘文件，这里是为存储的虚拟机磁盘文件进行命名，这里选择默认名字即可，单击【下一步】，如图 2-13 所示。

步骤 12: 准备创建虚拟机，默认即可，勾选【创建开启此虚拟机(P)】，然后单击【完成】，如图 2-14 所示。

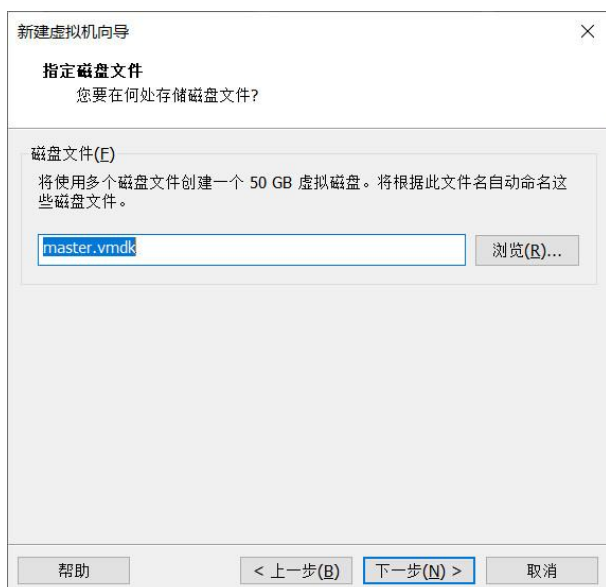


图 2-13 选择配置模式



图 2-14 准备创建虚拟机

步骤 13: 打开虚拟机之后，进入到 CentOS 的安装界面，单击进入虚拟机中，再次单击选择【Install CentOS 7】，然后按回车键，也可以选择不按回车，等待 60s 之后自动进入安装，如图 2-15 所示。

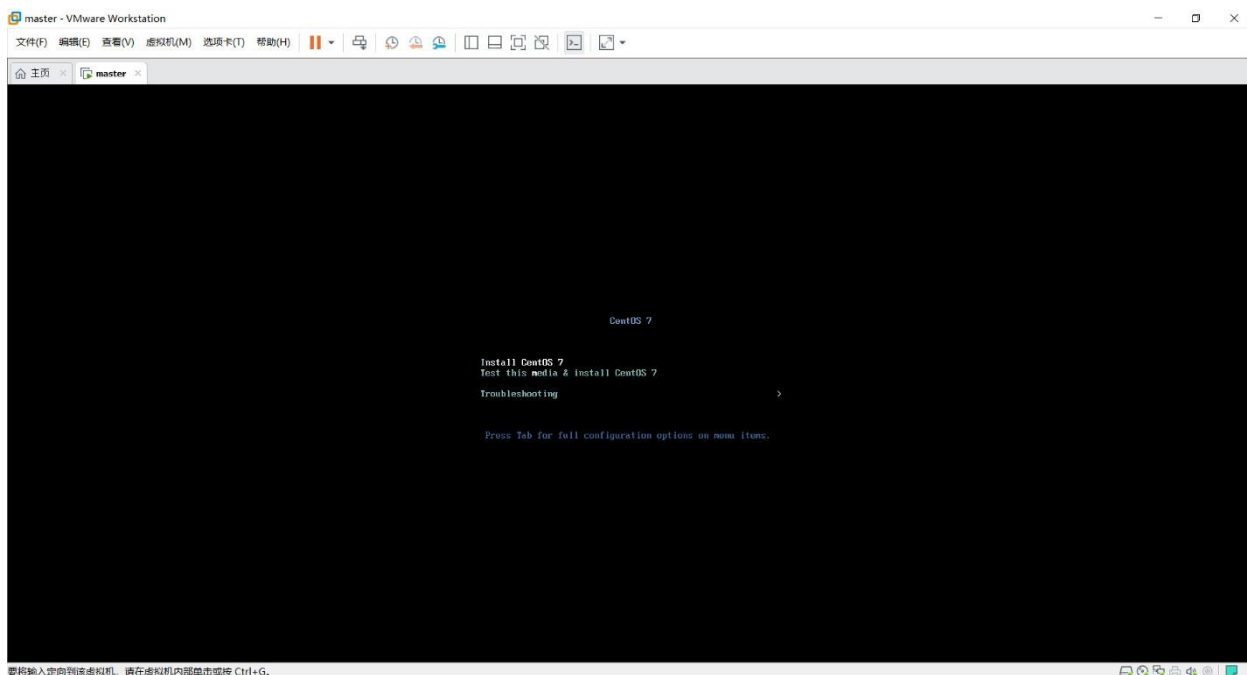


图 2-15 安装 CentOS 界面

步骤 14: 等待系统安装完成之后将自动进入 CentOS 7 语言配置界面, 如图 2-16 所示, 在左边选项卡中选择【中文】, 然后在右边选项卡中选择【简体中文(中国)】, 最后单击【继续(C)】按钮进入下一步。

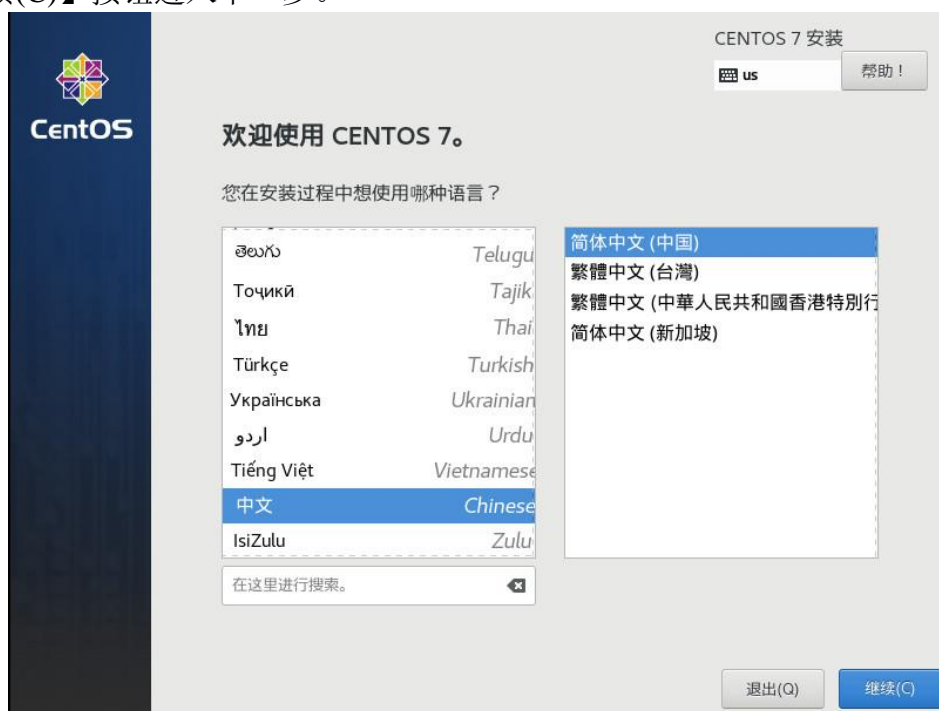


图 2-16 选择语言

步骤 15: 安装信息摘要, 选择第二栏【软件】中的【软件安装(S)】, 如图 2-17 所示, 在弹出的窗口中选择【GNOME 桌面】, 然后单击左上角【完成(D)】回到安装信息摘要界面, 如图 2-18 所示。



图 2-17 下载 Python3.7.2 安装包



图 2-18 GNOME 桌面

步骤 16: 接下来，在安装信息摘要第三栏【系统】中单击【安装位置(D)】，在弹出的安装目标位置窗口中直接单击左上角的【完成(D)】即可，如图 2-19、图 2-20 所示。



图 2-19 安装位置



图 2-20 选择磁盘

步骤 17: 返回安装信息摘要界面，单击【开始安装(B)】即可，如图 2-21 所示，进行系统最后的安装过程。



图 2-21 开始安装

步骤 18: 在安装过程中系统会提示用户设置，包括 ROOT 密码设置和创建普通用户，这里我们暂时不进行普通用户创建，单击【ROOT 密码】，如图 2-22 所示，进入 ROOT 密码设置窗口，本书所有 ROOT 密码均设置为“1”，输入设置密码之后，单击左上角的【完成(D)】返回，等待系统继续安装即可，如图 2-23 所示。



图 2-22 配置

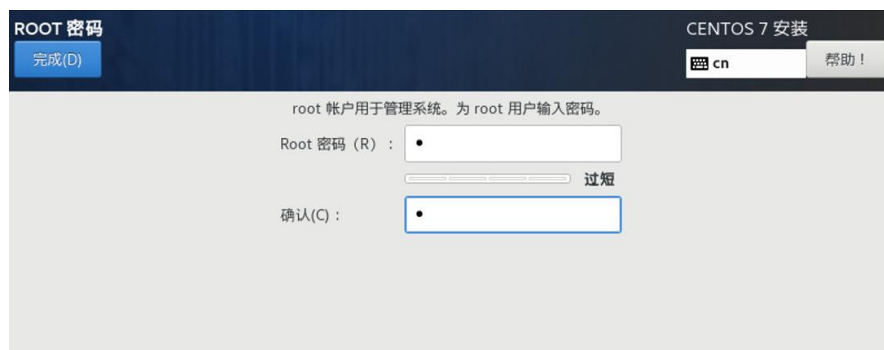


图 2-23 设置 root 密码

步骤 19: 等待完成安装之后，单击右下角的【重启(R)】，重启虚拟机，这时已经完成了 CentOS 7.9 的基本安装，如图 2-24 所示。



图 2-24 重启

步骤 20: 系统重启，等待完成安装之后，会提示我们进行简单的初始化设置，例如【LICENSING 许可授权】，如图 2-25 所示，我们单击进入设置窗口，勾选【我同意许可协议】，并单击左上角【完成(D)】返回首页，如图 2-26 所示。



图 2-25 LICENSING 许可授权

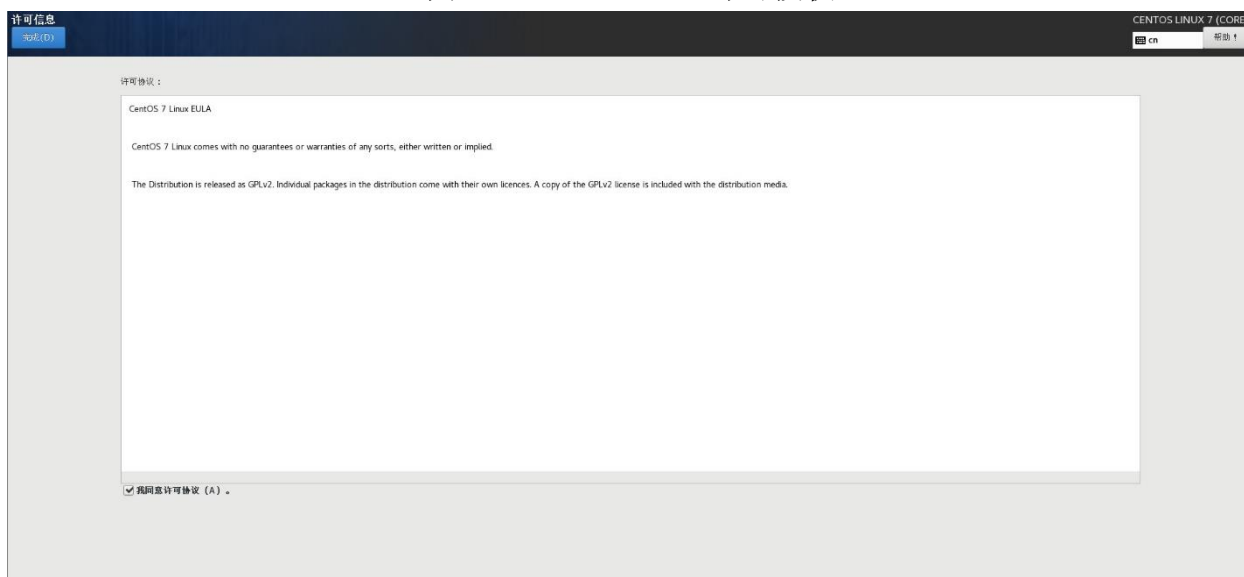


图 2-26 完成

步骤 21：接下来是一些常规的系统初始化配置，分别是语言选择、输入键盘选择、隐私、在线账号，这里按照系统提示选择默认选项即可，如图 2-27 至图 2-30 所示。值得注意的是，在以上四步完成之后，进入到用户设置页面，本书统一用户名设置为“bigdata”，用户密码设置为“BD123456?”，如图 2-31 和图 2-32 所示，读者也可以根据自己的需要设置满足条件的用户名或密码。

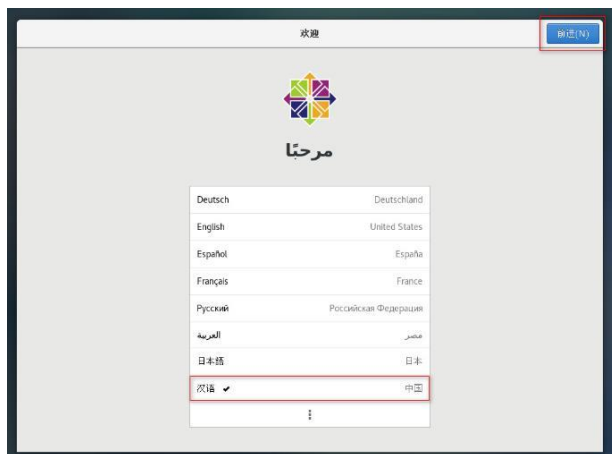


图 2-27 语言选择



图 2-28 输入设置



图 2-29 隐私设置



图 2-30 连接在线帐号



图 2-31 普通用户设置



图 2-32 普通用户密码设置

步骤 22: 完成上述所有配置之后，我们就正式开始了 CentOS Linux 之旅啦！由于我们要搭建的 Hadoop 完全分布式集群由三台机器组成，分别是 master、slave1、slave2，所以我们需要对 Linux 的主机名进行设置，单击桌面左上角的【应用程序】，在弹出的下拉菜单中选择【系统工具】，并在右侧弹出的选型卡中选择【设置】按钮并单击，进入到设置窗口，在设置窗口的最下面选择【设备信息】，如图 2-33 所示。

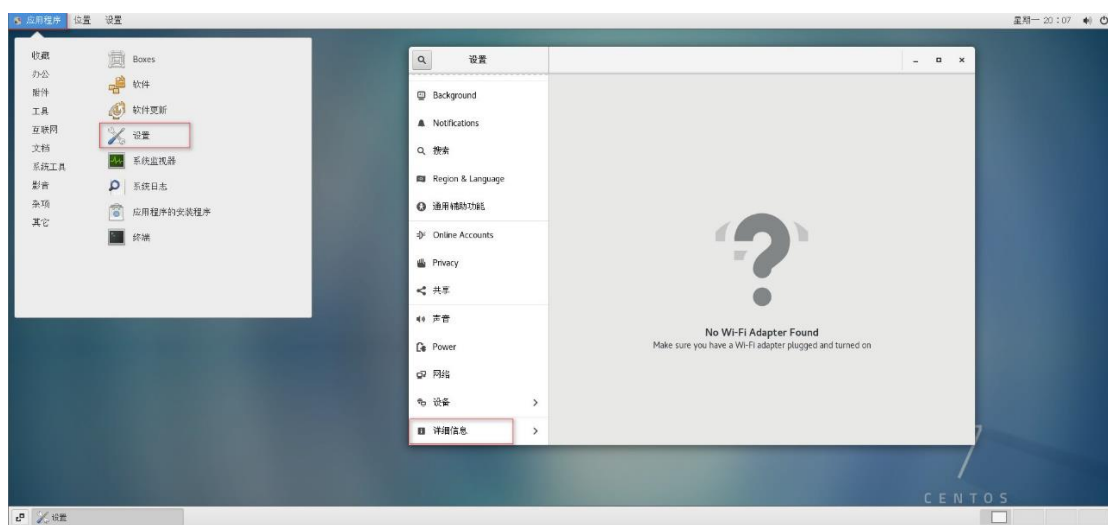


图 2-33 设置页面

步骤 23: 在设备信息窗口中切换到【About】选项卡，在右侧【设备名称】中输入“master”即可完成 Linux 主机名的设置工作，如图 2-34 所示。

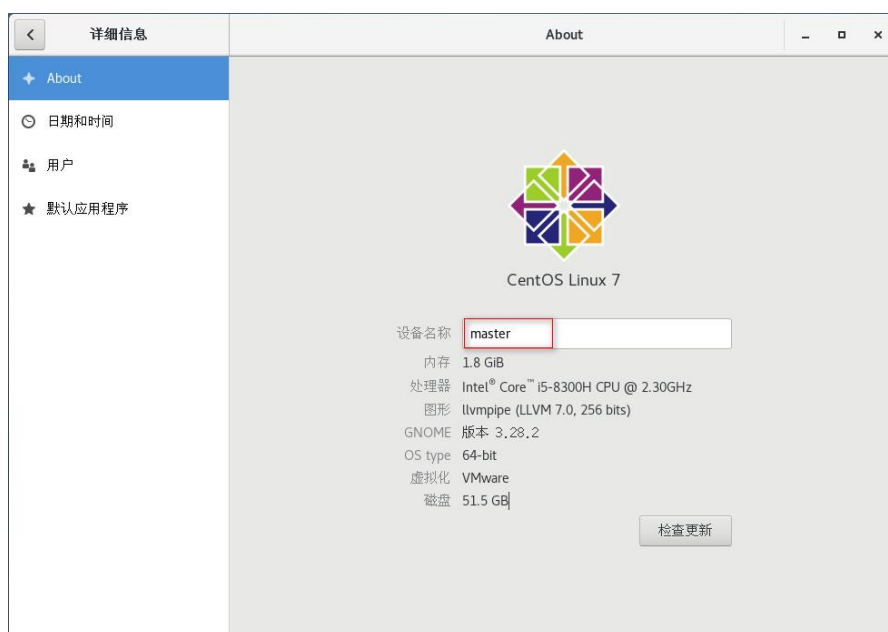


图 2-34 About 设置主机名

温馨提示：

2.2.3 Linux 设置固定 IP

本书搭建的集群包括 3 台虚拟机，每台虚拟机均使用 NAT 模式连接网络，通过为其分配固定 IP 地址，可以保证每台虚拟机的 IP 处于同一网段内。本小节以 master 为例，详细介绍如何配置虚拟机固定 IP 地址，主要步骤如下。

步骤 01：查看虚拟机所处子网网段。保证 3 台虚拟机能够处于同一子网网段内是集群搭建的关键步骤。首先，在 VMware 菜单栏单击【编辑】，在弹出的下拉菜单中单击【虚拟网络编辑器】，并在弹出的窗口中单击选中【VMnet8】，VMnet8 即为虚拟机 NAT 模式的网络信息，查看其所处的子网地址，如本书中虚拟机所处的子网地址为“192.168.75.0”，所以虚拟机子网网段为“75”网段，如图 2-35、图 2-36 所示。

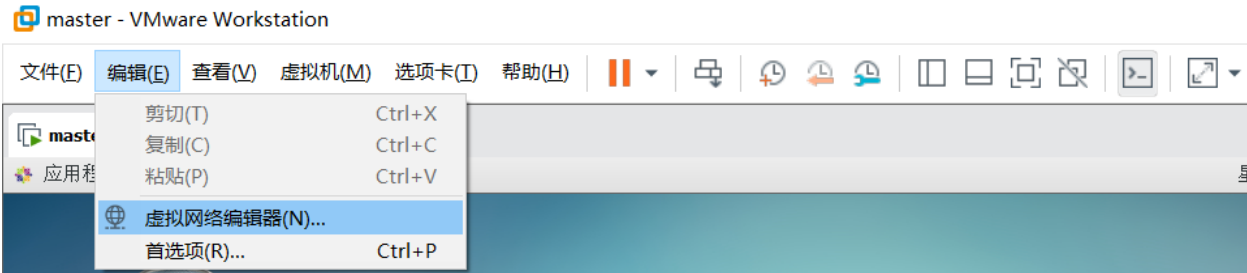


图 2-35 虚拟网络编辑器



图 2-36 NAT 模式

步骤 02: 关闭虚拟机防火墙，包括防火墙、内核防火墙。这里需要通过命令进行设置，在 Linux 中命令通过终端的方式运行（类似 Windows 中的 cmd），打开终端的方法：在系统桌面单击鼠标右键，在弹出的菜单中单击【打开终端】，如图 2-37 所示。

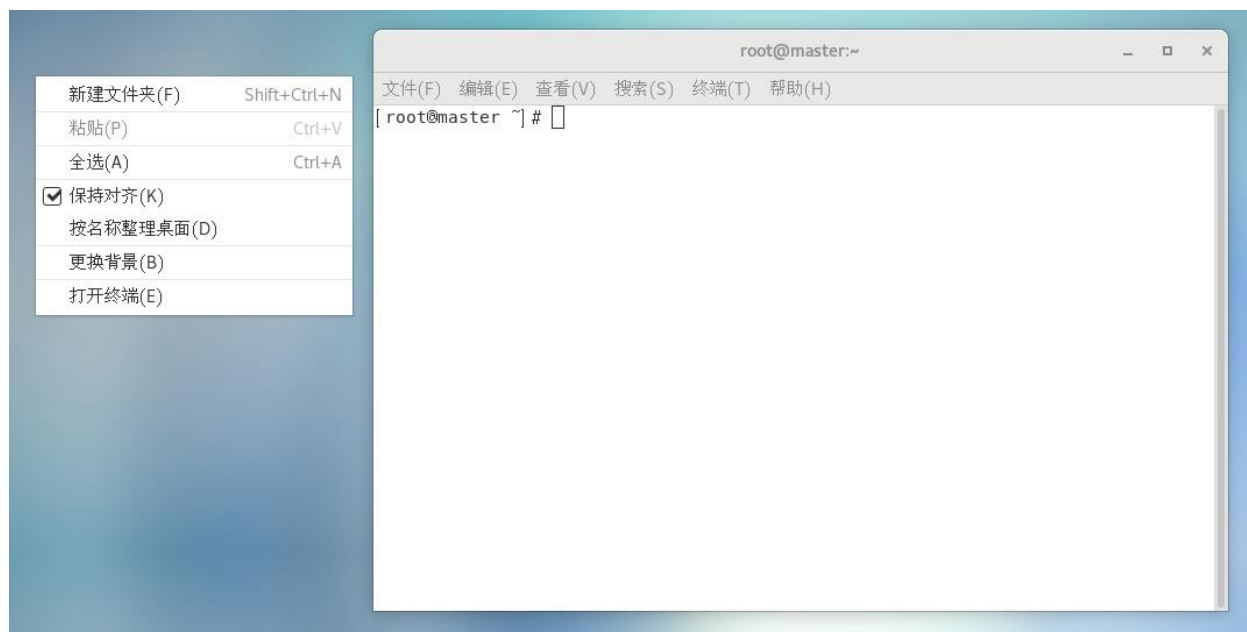


图 2-37 打开终端

关闭防火墙代码如下。

```
# 直接关闭防火墙
systemctl stop firewalld.service
# 禁止 firewall 开机启动
systemctl disable firewalld.service
# 查看防火墙状态：
firewall-cmd --state
# 临时关闭内核防火墙
setenforce 0
```

运行结果如图 2-38 所示。



图 2-38 关闭防火墙和内核防火墙

永久关闭防火墙需要通过编辑配置文件，编辑配置文件命令如下。

```
# 永久关闭内核防火墙
vim /etc/selinux/config
```

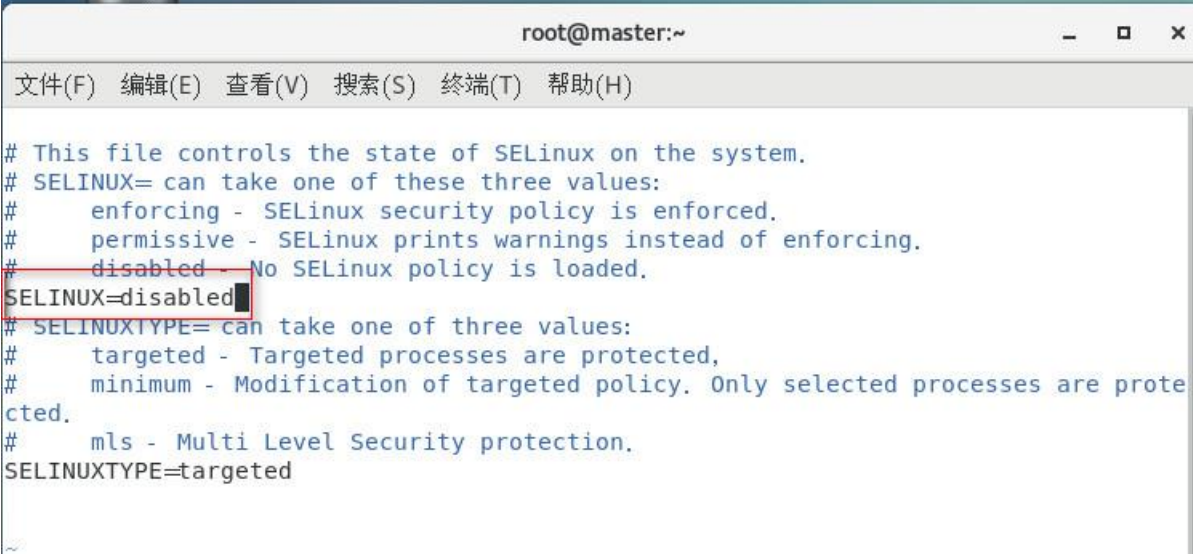
输出上述命令后，会自动打开内核防火墙配置文件，找到如下内容。

SELINUX=enforcing

将其改成如下。

SELINUX=disabled

修改结果如图 2-39 所示。



```
root@master:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
# This file controls the state of SELinux on the system.  
# SELINUX= can take one of these three values:  
#   enforcing - SELinux security policy is enforced.  
#   permissive - SELinux prints warnings instead of enforcing.  
#   disabled - No SELinux policy is loaded.  
SELINUX=disabled  
# SELINUXTYPE= can take one of three values:  
#   targeted - Targeted processes are protected,  
#   minimum - Modification of targeted policy. Only selected processes are protected.  
#   mls - Multi Level Security protection.  
SELINUXTYPE=targeted
```

图 2-39 config 文件展示

温馨提示：

Vim 是 Linux 中的文本编辑工具。上述内容使用 Vim 修改方法：输入“vim 文件名”进入文本之后，在英文输入状态下输入字母【i】即可进入文

本编辑状态。此时可以通过方向键定位到需要修改的位置，并进行修改。

步骤 03：修改主机名，通过修改配置文件完成主机名的修改代码如下。

vim /etc/sysconfig/network

加入以下内容

NETWORKING=yes

HOSTNAME=master

修改结果如图 2-40 所示。



```
root@master:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[ root@master ~] # vim /etc/sysconfig/network  
[ root@master ~] # cat /etc/sysconfig/network  
# Created by anaconda  
NETWORKING=yes  
HOSTNAME=master  
[ root@master ~] #
```

图 2-40 network

步骤 04：设置虚拟机固定 IP。由于本书中虚拟机所处子网地址为“192.168.75.0”，所以读者需要根据自己虚拟机所处的网段进行相关的修改，修改固定 IP 配置文件代码如下。

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

修改后 ifcfg-ens33 文件内容如下所示。

```
TYPE=Ethernet
BROWSER_ONLY=no
DEFROUTE=yes
NAME=ens33
DEVICE=ens33
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.75.100
NETMASK=255.255.255.0
GATEWAY=192.168.75.1
DNS1=114.114.114.114
DNS2=8.8.8.8
```

其中，DEVICE 是指设备名，ONBOOT 是设置系统启动时是否激活网卡，BOOTPROTO 的值可以设置为 dhcp、none、bootp 或 static，它们分别代表的含义如表 2-2 所示。

表 2-2 设置 BOOTPROTO 的值

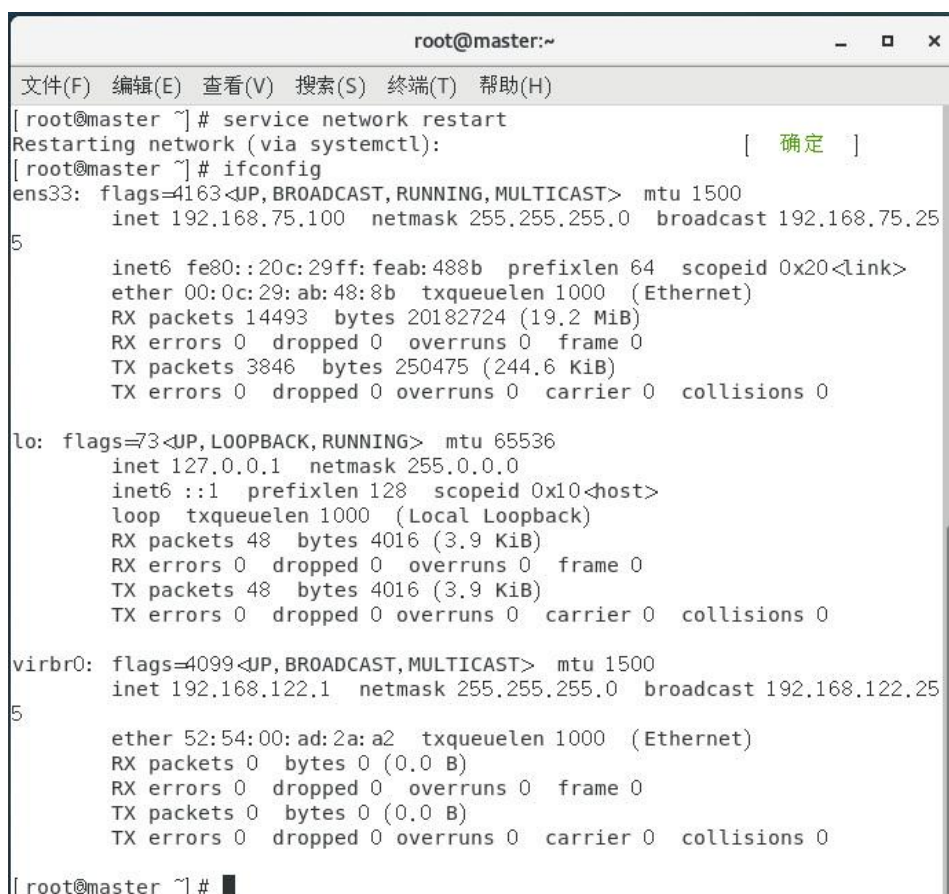
| | |
|--------|-----------------------------|
| dhcp | 设置网卡绑定的时候通过 DHCP 协议的方法来获得地址 |
| none | 设置网卡绑定的时候不适用任何协议 |
| bootp | 设置网卡绑定的时候使用 BOOTP 协议 |
| static | 设置网卡绑定的时候使用静态协议 |

IPADDR 前面部分是虚拟机所处的子网网段，即“192.168.75”，最后一位“100”可以理解为虚拟机在该子网内的端口地址（读者可以灵活设置未被占用的端口），GATEWAY 是网关，需要和子网网段保持一致，此时为宿主机（即主机虚拟 IP）最后一位端口为“1”，DNS1 和 DNS2 按着本书中的设置即可。

在完成上述配置文件的修改后通过重启网络时新的 IP 地址生效，重启网络命令如下。

```
/etc/init.d/network restart
# 或
service network restart
```

在成功重启网络后可以通过【ifconfig】命令查看修改后的 IP 地址，如图 2-41 所示。



```
root@master:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@master ~]# service network restart  
Restarting network (via systemctl): [ 确定 ]  
[root@master ~]# ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.75.100 netmask 255.255.255.0 broadcast 192.168.75.255  
    inet6 fe80::20c:29ff:feab:488b prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:ab:48:8b txqueuelen 1000 (Ethernet)  
    RX packets 14493 bytes 20182724 (19.2 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3846 bytes 250475 (244.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 48 bytes 4016 (3.9 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 48 bytes 4016 (3.9 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255  
    ether 52:54:00:ad:2a:a2 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
[root@master ~]#
```

图 2-41 ip 地址配置

通过以上步骤就完成了 Linux 固定 IP 的设置工作。

温馨提示：

有些虚拟机中图形界面存在两套管理 ip 的工具，需要禁止一下，否则可能出现重启网络失败或者不生效，禁用命令如下。

```
service NetworkManager stop
```

2.3 Java 软件安装

JDK 是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的 Java 应用程序。JDK 是整个 Java 的核心，它包含了 Java 的运行环境，Java 工具和 Java 基础的类库。而 Hadoop 也是基于 Java 语言开发的，所以 Java JDK 软件也是 Hadoop 软件运行必备的环境依赖，本节将分别介绍在 Windows 下和 Linux 下如何安装 Java JDK，并通过版本验证的方法验证 Java 是否安装成功。

2.3.1 Windows 安装 JDK

JDK 的下载可以通过 Oracle 官网进行选择和下载，也可以使用本书配套的软件包。JDK 下载官网为：<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>。具体下载步骤如下。

步骤 01：双击 jdk 安装包 jdk-8u192-windows-x64.exe，如图 2-42 所示，并单击【下一步】按钮。



图 2-42 jdk 安装

步骤 02：默认安装目录在 C 盘，可以点击【更改】按钮，自定义安装目录（不建议选择含有中文字符的路径），如图 2-43 所示，然后单击【下一步】，等待程序安装完成即可。



图 2-43 默认安装

步骤 03: 在完成 JDK 安装之后，系统会弹出安装 JRE 的提示窗口，这里和步骤 02 相似，可以单击【更改】按钮，选择自定义的安装目录，然后单击【下一步】继续安装即可，如图 2-44 所示。等待完成安装之后，单击窗口中的【关闭】，即可完成 JDK 的安装，如图 2-45 所示。



图 2-44 下一步



图 2-45 等待安装完成

步骤 04: 配置环境变量。在完成 JDK 安装之后需要为其配置环境变量。首先，右键单击【计算机】，单击选择【属性】选项，在弹出的系统设置窗口中选择【高级系统设置】，由此进入到【系统属性】对话框，单击【环境变量】按钮，在弹出的【环境变量】窗口中进行相关配置，如图 2-46 所示。

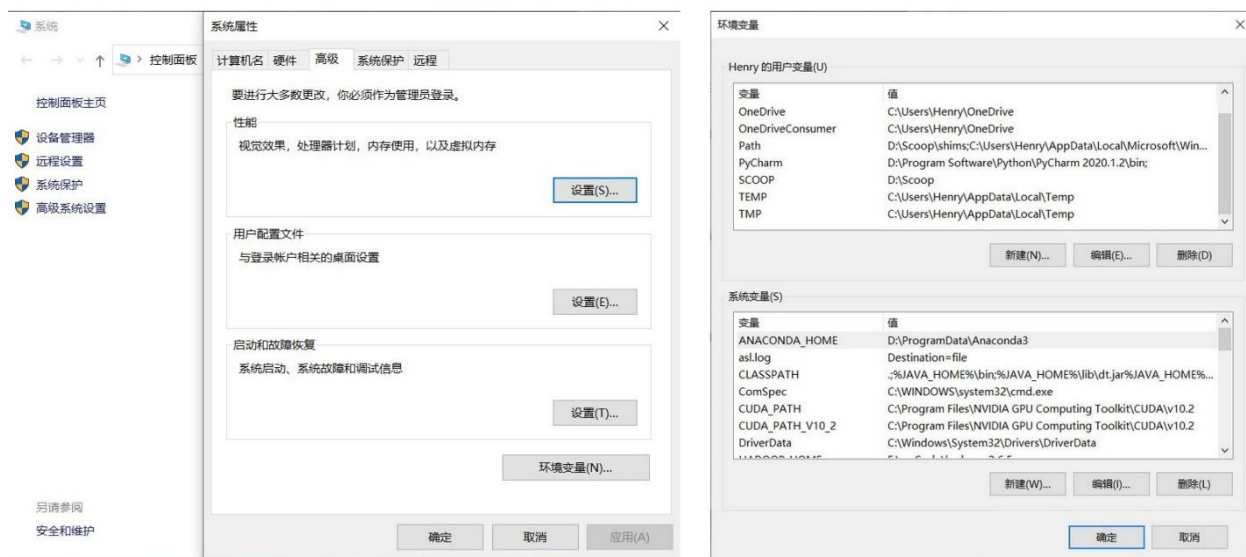


图 2-46 设置环境变量

步骤 04: 在【系统变量】部分，单击【新建】，创建“JAVA_HOME”变量，变量值为安装 JDK 时的路径，如图 2-47 所示。接下来，再次单击【新建】按钮，创建“CLASSPATH”变量，变量值输入“.;%JAVA_HOME%\jre\lib\rt.jar;.”（标点符号为英文状态下），如图 2-48 所示。

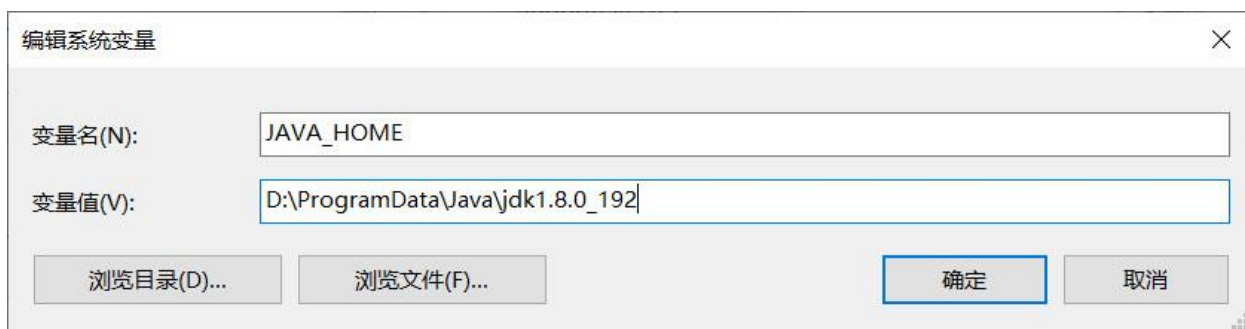


图 2-47 配置“JAVA_HOME”环境变量

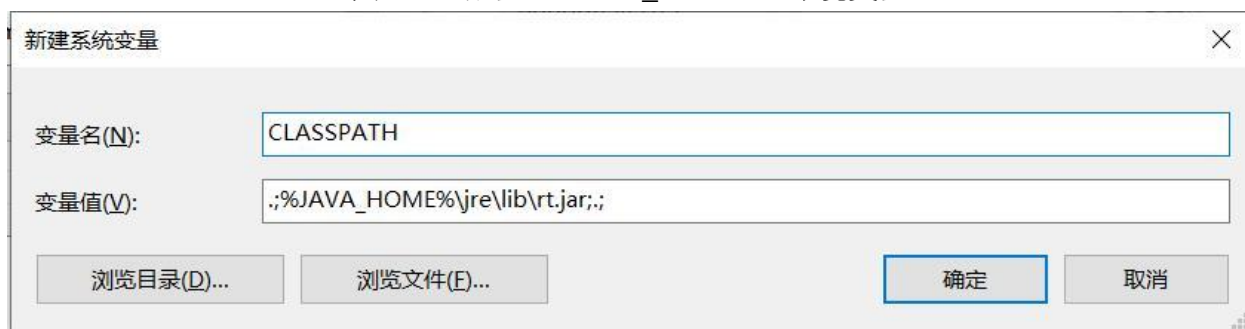


图 2-48 配置“CLASSPATH”环境变量

步骤 05: 接下来, 在【系统变量】中找到“PATH”变量, 双击打开, 在变量值中添加“;%JAVA_HOME%\bin;”（最前面有一个“;”符号）, 如图 2-49 所示。以上完成之后, 点击【确定】关闭环境变量配置的窗口即可。

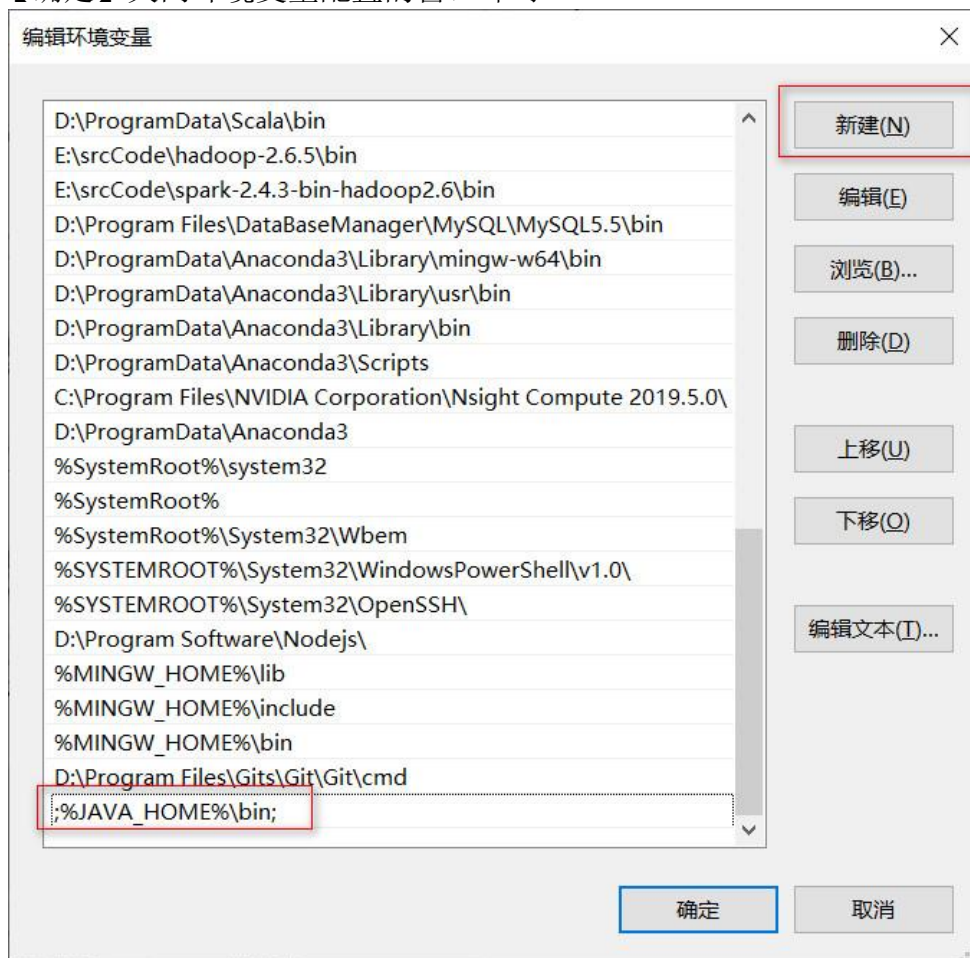


图 2-49 配置环境变量完毕

步骤 06: 最后测试环境变量配置是否正确。按下键盘【Win+R】，在弹出的运行窗口中输入“cmd”然后按回车键，如图 2-50 所示。进入 cmd 命令窗口，在其中输入“java -version”，若弹出如图 2-51 所示的信息，则代表环境变量配置成功。

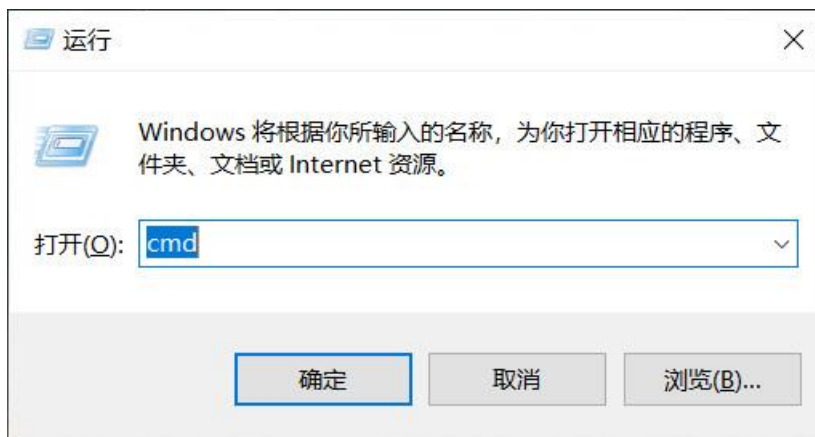


图 2-50 打开 cmd 命令行工具



图 2-51 jdk 安装成功验证

2.3.2 Linux 安装 JDK

Linux 安装 JDK 方法如下所示。

步骤 01: 由于系统设置语言为中文，为了使用方便，我们将常用的目录，如“桌面”、“下载”等，改回英文状态，在虚拟机桌面空白处右键单击，选择【打开终端】，执行代码如代码如下所示。

```
export LANG=en_US           # 将系统语系改为英文
xdg-user-dirs-gtk-update    # 弹出窗口,根据选择,更新目录
export LANG=zh_CN.UTF-8     # 再将语系改为中文
```

运行结果如图 2-52 所示。完成修改之后重启一下系统即可。

```
root@master:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@master ~] # export LANG=en_US  
[root@master ~] # xdg-user-dirs-gtk-update  
Gtk-Message: 11:42:07.705: GtkDialog mapped without a transient parent. This is discouraged.  
Moving DESKTOP directory from 桌面 to Desktop  
Moving DOWNLOAD directory from 下载 to Downloads  
Moving TEMPLATES directory from 模板 to Templates  
Moving PUBLICSHARE directory from 公共 to Public  
Moving DOCUMENTS directory from 文档 to Documents  
Moving MUSIC directory from 音乐 to Music  
Moving PICTURES directory from 图片 to Pictures  
Moving VIDEOS directory from 视频 to Videos  
[root@master ~] # export LANG=zh_CN.UTF-8  
[root@master ~] # ls  
anaconda-ks.cfg  Documents  initial-setup-ks.cfg  Pictures  Templates  
Desktop          Downloads  Music                 Public    Videos  
[root@master ~] #
```

图 2-52 运行结果

步骤 02: 删除 CentOS 自带的 JDK，执行如下命令删除系统自带版本的 JDK。

查看 java 的安装位置

rpm -qa | grep java

运行结果如图 2-53 所示。

```
root@master:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
[root@master ~] # rpm -qa | grep java  
java-1.7.0-openjdk-headless-1.7.0.261-2.6.22.2.el7_8.x86_64  
python-javapackages-3.4.1-11.el7.noarch  
tzdata-java-2020a-1.el7.noarch  
java-1.8.0-openjdk-headless-1.8.0.262.b10-1.el7.x86_64
```

图 2-53 查看 java 安装

通过使用“rpm -e --nodeps package_name”逐一删除（注意，其中 package_name 指代图 2-53 中运行结果中展示的 java 安装包名称），以本书为例，执行代码如下。

删除 java 安装包

rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.261-2.6.22.2.el7_8.x86_64

rpm -e --nodeps python-javapackages-3.4.1-11.el7.noarch

rpm -e --nodeps tzdata-java-2020a-1.el7.noarch

rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.262.b10-1.el7.x86_64rpm

再次查看是否删除完全

rpm -qa | grep java

运行结果如图 2-54 所示。



图 2-54 再次查看 java

此时，说明系统自带 JDK 已经完全删除。

步骤 03：下载并安装 JDK。下载地址可参考本书提供的 Linux JDK 安装包，也可以通过命令的方式在 Linux 中直接下载，JDK 下载地址为：

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>。通过拖拽的方式将下载好的 JDK 安装包 jdk-8u172-linux-x64.tar.gz 上传到 master 虚拟机中，如图 2-55 所示，也可以选择其他 FTP 工具，例如 Xftp 工具软件。

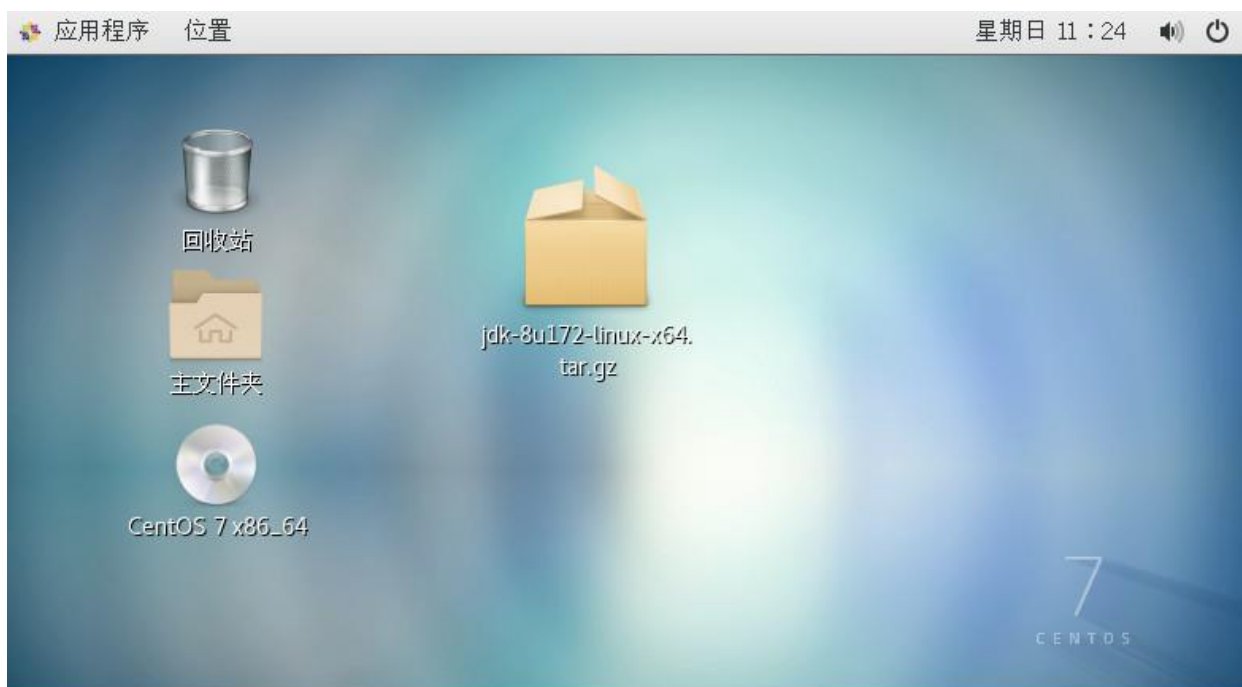


图 2-55 Linux jdk 示意图

步骤 04：在终端中执行如下命令，将安装包移动到/usr/local/src 目录中。

```
cd Desktop/
mv jdk-8u172-linux-x64.tar.gz /usr/local/src/
cd /usr/local/src/
ls # 查看是否移动成功
```

运行结果如图 2-56 所示。



```
root@master:/usr/local/src
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master ~] # cd Desktop/
[root@master Desktop] # ls
jdk-8u172-linux-x64.tar.gz
[root@master Desktop] # mv jdk-8u172-linux-x64.tar.gz /usr/local/src/
[root@master Desktop] # ls
[root@master Desktop] # cd /usr/local/src/
[root@master src] # ls
jdk-8u172-linux-x64.tar.gz
[root@master src] #
```

图 2-56 移动 jdk 结果展示

步骤 05: 安装配置 JDK, 终端中执行如下命令, 解压 JDK 安装包。

```
cd /usr/local/src/
tar -zxvf jdk-8u172-linux-x64.tar.gz
```

```
# 查看是否解压成功
ls
```

运行结果如图 2-57 所示。



```
root@master:/usr/local/src
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master src] # ls
jdk1.8.0_172 jdk-8u172-linux-x64.tar.gz
[root@master src] #
```

图 2-57 解压 jdk 结果展示

同样的, 类似于 Windows 下, 我们需要对 JDK 进行环境变量配置, 执行代码如下。

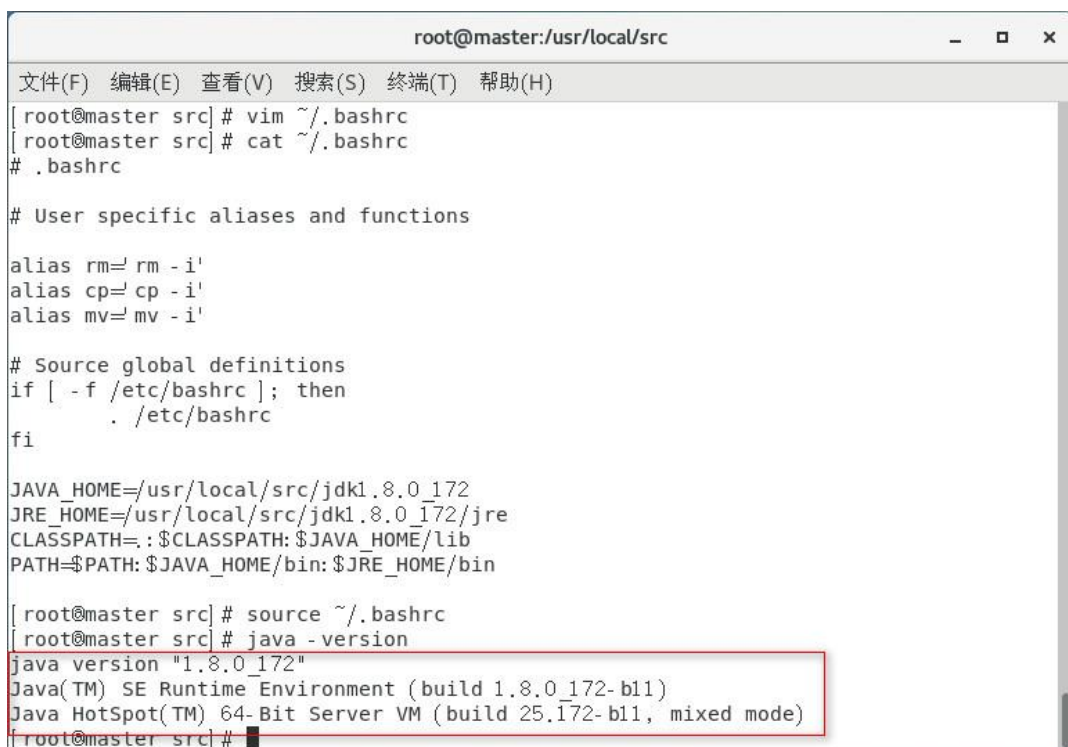
```
vim ~/.bashrc

# 在最下面一行加入如下内容
JAVA_HOME=/usr/local/src/jdk1.8.0_172
JRE_HOME=/usr/local/src/jdk1.8.0_172/jre
CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin

# 退出后刷新环境变量
source ~/.bashrc

# 查看 java 是否安装成功
Java -version
```

运行结果如图 2-58 所示。



```
root@master:/usr/local/src
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master src] # vim ~/.bashrc
[root@master src] # cat ~/.bashrc
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

JAVA_HOME=/usr/local/src/jdk1.8.0_172
JRE_HOME=/usr/local/src/jdk1.8.0_172/jre
CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin

[root@master src] # source ~/.bashrc
[root@master src] # java -version
java version "1.8.0_172"
Java(TM) SE Runtime Environment (build 1.8.0_172-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.172-b11, mixed mode)
[root@master src] #
```

图 2-58 测试 Linux jdk 安装

至此，Linux 中安装 JDK 的内容已经完成。

2.4 Hadoop 完全分布式集群搭建

学习大数据 Hadoop 框架，搭建 Hadoop 集群是必要的环境，是对大数据有个详细认知的基础。本节任务是搭建 Hadoop 的完全分布式环境，包括一个 master 主节点和两个 slave 子节点。

2.4.1 安装 Hadoop

首先，安装 Hadoop 和安装 Java 软件都需要将安装包上传到虚拟机 Linux 中，方法同 1.3.2 节中介绍的。在上传完成之后，同样将安装包移动到/usr/local/src 目录下，然后执行“tar -zxvf hadoop-2.6.5.tar.gz”将安装包进行解压，如图 2-59 所示。



```
root@master:/usr/local/src
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master src] # ls
hadoop-2.6.5  hadoop-2.6.5.tar.gz  jdk1.8.0_172  jdk-8u172-linux-x64.tar.gz
[root@master src] #
```

图 2-59 解压 hadoop 安装包

Hadoop 安装相对 Java 安装较为复杂，Hadoop 安装总共涉及的配置文件有：core-site.xml、mapred-site.xml、yarn-site.xml、hdfs-site.xml、hadoop-env.sh、yarn-env.sh、slaves。上述配置文件均在/usr/local/src/hadoop-2.6.5/etc/hadoop 目录下（/usr/local/src 为

Hadoop 安装包解压的根目录），进入该目录，逐次对上述配置文件进行修改配置。如图 2-60 所示。

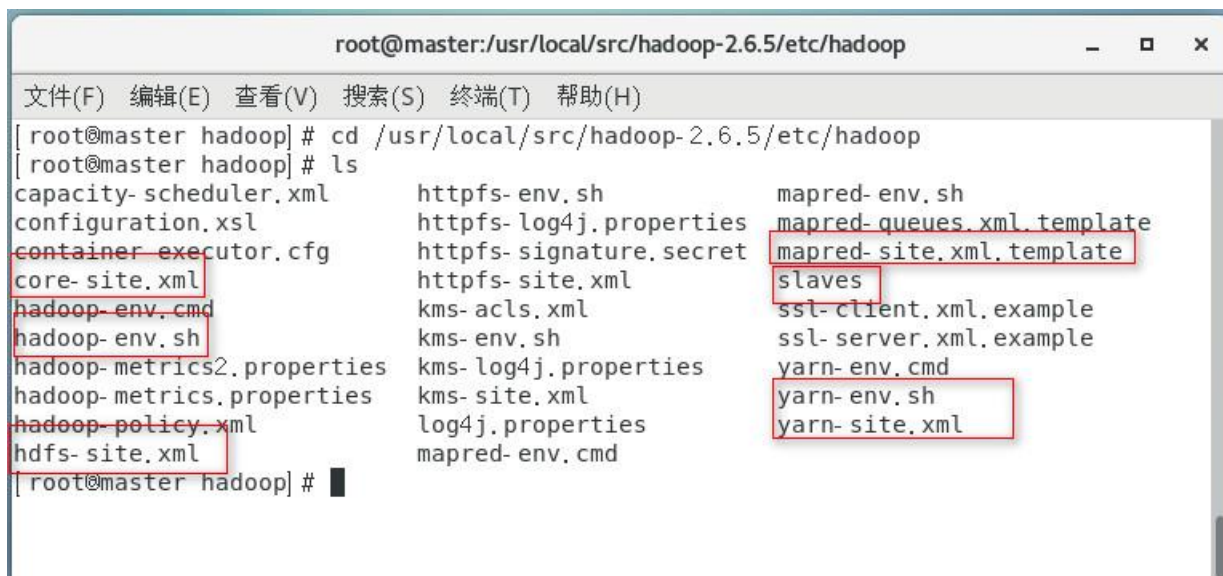


图 2-60 展示 xml 文件

修改详情如下所示。

1. 修改 core-site.xml 文件

core-site.xml 是整个 Hadoop 中比较核心的配置文件，我们所需要配置的内容包括 Hadoop 的 HDFS 系统地址，以及 Hadoop 临时文件的存储路径。这里我们介绍另外一种编辑 Linux 文件的方法，在终端中进入到该目录后执行“gedit core-site.xml”，系统会自动打开文本编辑器，类似于 Windows 中的记事本工具，可以方便的使用鼠标进行文本位置定位，以及粘贴复制等功能，如图 2-61 所示。

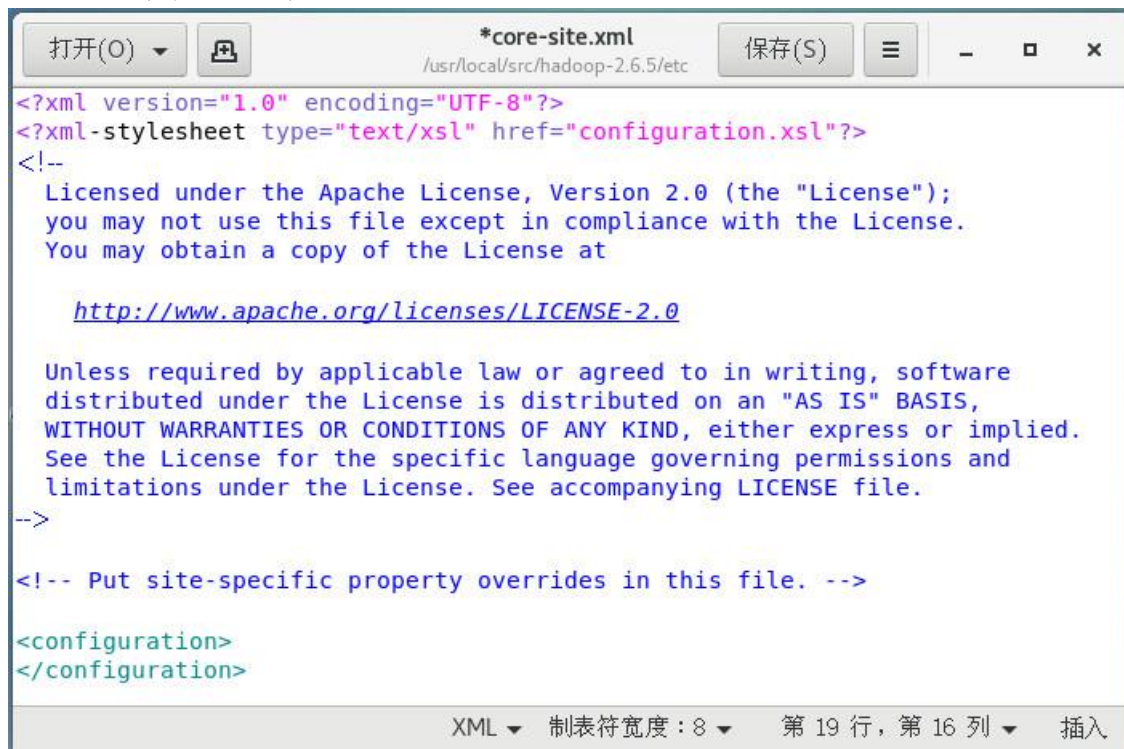


图 2-61 core-site.xml 配置

在 core-site.xml 中添加的代码内容如下所示。

```
<configuration>
```



```

<property>
  <name>fs.defaultFS</name>
  <value>hdfs://192.168.75.100:9000</value> # 或者 hdfs://master:9000
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/usr/local/src/hadoop-2.6.5/tmp</value>
</property>
</configuration>

```

2. 修改 mapred-site.xml 文件

mapred-site.xml 是 Hadoop 中 MapReduce 相关的配置文件，这里配置的包括三个内容：因为 Hadoop 2.x 引入了 Yarn 框架，所以在该配置文件中需要指定 MapReduce 使用的框架为 Yarn，以及 MapReduce 相关的 JobHistoryServer 的相关配置（JobHistoryServer 是为了记录 MapReduce 任务运行的日志服务）。这里不同于 core-site.xml 配置的地方，Hadoop 安装包中只提供了 mapred-site.xml.template 模板文件，我们需要执行“cp mapred-site.xml.template mapred-site.xml”命令，从而得到 mapred-site.xml 文件，代码如下所示。

```

<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>

```

3. 修改 yarn-site.xml 文件

yarn-site.xml 是关于 Yarn 框架的相关配置，该文件中配置的内容较多，均是关于 Yarn 框架的如 Yarn 服务、调度服务等地址端口信息，配置代码如下所示。

```

<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8032</value>
  </property>

```

```

</property>
<property>
  <name>yarn.resourcemanager.scheduler.address</name>
  <value>master:8030</value>
</property>
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>master:8035</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>
</configuration>

```

4. 修改 hdfs-site.xml 文件

hdfs-site.xml 是关于 Hadoop 中存储系统 HDFS 相关配置的文件，主要包括 5 个内容：NameNode、元数据、DataNode 数据存储位置、SecondaryNameNode 地址以及 HDFS 存储的副本数，默认是 3 个副本，可以根据实际情况进行修改，修改内容如下所示。

```

<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///usr/local/src/hadoop-2.6.5/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///usr/local/src/hadoop-2.6.5/dfs/data</value>
  </property>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>master:9001</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>

```

5. 修改 hadoop-env.sh 文件

hadoop-env.sh 文件是配置 Hadoop 运行相关的基本环境，这里只需要修改 JDK 的实际路径即可，找到文件中 JAVA_HOME 配置的位置修改，修改内容如下所示。

| | |
|--|--------|
| # The java implementation to use. | # 原始内容 |
| export JAVA_HOME=/usr/local/src/jdk1.8.0_172 | # 修改内容 |

6. 修改 yarn-env.sh 文件

yarn-env.sh 文件的作用类似于 hadoop-env.sh，它提供了 Yarn 框架运行的配置，同样，这里也只需要修改 JDK 的实际路径，修改内容如下所示。

```
# some Java parameters
# export JAVA_HOME=/home/y/libexec/jdk1.6.0/           # 原始内容
export JAVA_HOME=/usr/local/src/jdk1.8.0_172</configuration> # 修改内容
```

7. 修改 slaves 文件

slaves 文件中配置了所有 slave 节点的信息，即 slave 节点的主机名，如下所示。

```
slave1
slave2
```

上述 Hadoop 的所有相关配置已经完成，除了 Hadoop 相关的配置文件之外，为了保证集群各个节点之间能够正常通信，还需要修改 hosts 文件，hosts 文件位于“/etc/”路径下，可以使用命令“gedit /etc/hosts”打开 hosts 文件进行修改，修改内容如下所示。

```
192.168.75.100 master
192.168.75.101 slave1
192.168.75.102 slave2
```

最后，配置 Hadoop 环境变量，执行命令“vim ~/.bashrc”，在文件末尾添加 Hadoop 环境变量如图 2-62 所示。在添加完 Hadoop 环境变量之后，在终端执行命令“source ~/.bashrc”刷新环境变量以生效。

```
# ~/.bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

export JAVA_HOME=/usr/local/src/jdk1.8.0_172
export JRE_HOME=/usr/local/src/jdk1.8.0_172/jre
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin

export HADOOP_HOME=/usr/local/src/hadoop-2.6.5
export PATH=$PATH:$HADOOP_HOME/bin
```

图 2-62 修改环境变量

2.4.2 克隆虚拟机

在 1.3.1 小节中完成了 master 节点 Hadoop 的相关配置，由于集群包含 master、slave1、slave2 节点，本小节中将介绍通过克隆虚拟机的方式快速创建 slave1 和 slave2 节点。需要注意的是，克隆需要在虚拟机处于关机状态下进行，所以依次单击 VMware 软件菜单中的【虚拟机】—>【电源】—>【关机】，在将 master 节点关机之后，下面详细介绍虚拟机的克隆方法。

步骤 01：在 VMware 软件 master 界面，右键单击，选择【管理】——>【克隆】选项，如图 2-63 所示。在弹出的克隆虚拟机向导单击【下一页】按钮。

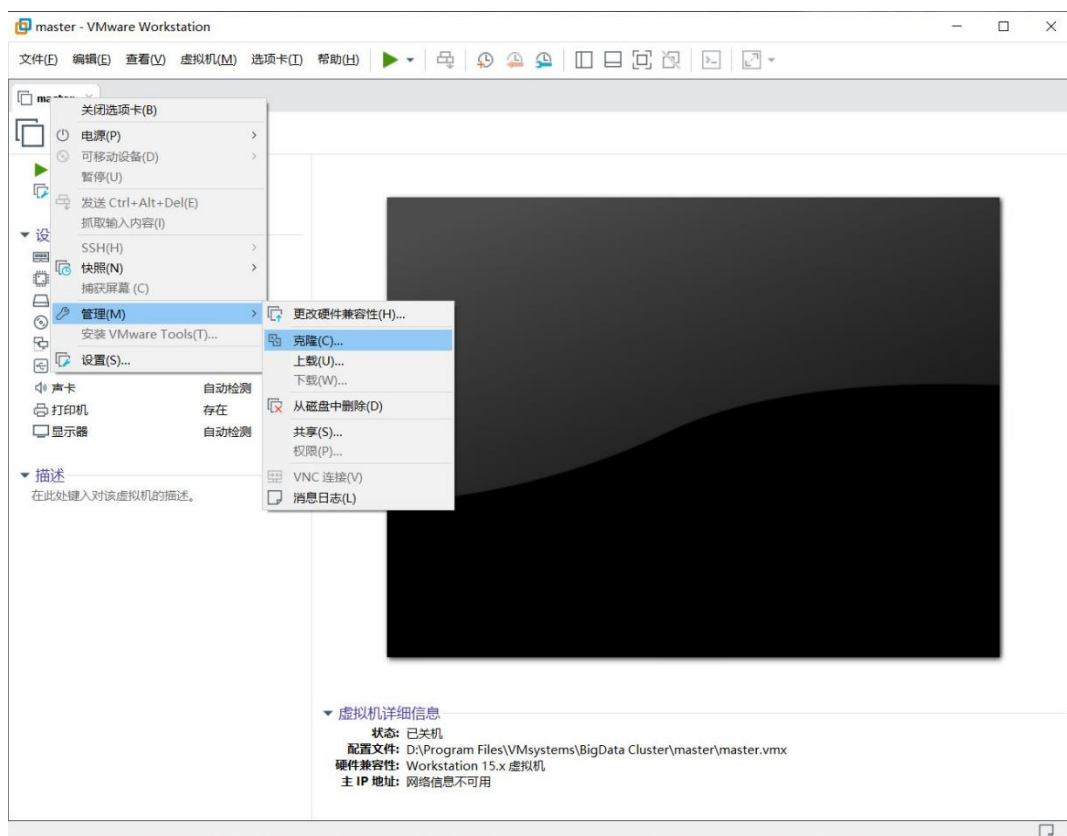


图 2-63 【管理】克隆虚拟机

步骤 02：在克隆虚拟机向导界面中，选中【虚拟机中的当前状态（C）】选项，然后继续单击【下一页】按钮，如图 2-64 所示。

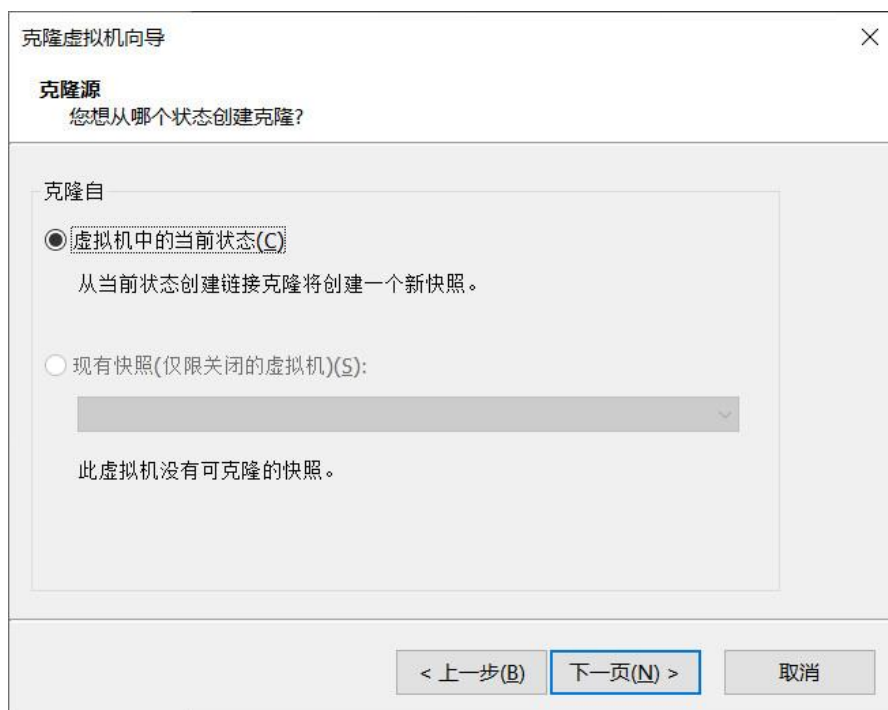


图 2-64 克隆虚拟机向导

步骤 03: 在克隆类型页面, 选中【创建完整克隆 (F)】(注意此处默认是【创建链接克隆 (L)】), 单击【下一页 (N)】按钮, 如图 2-65 所示。

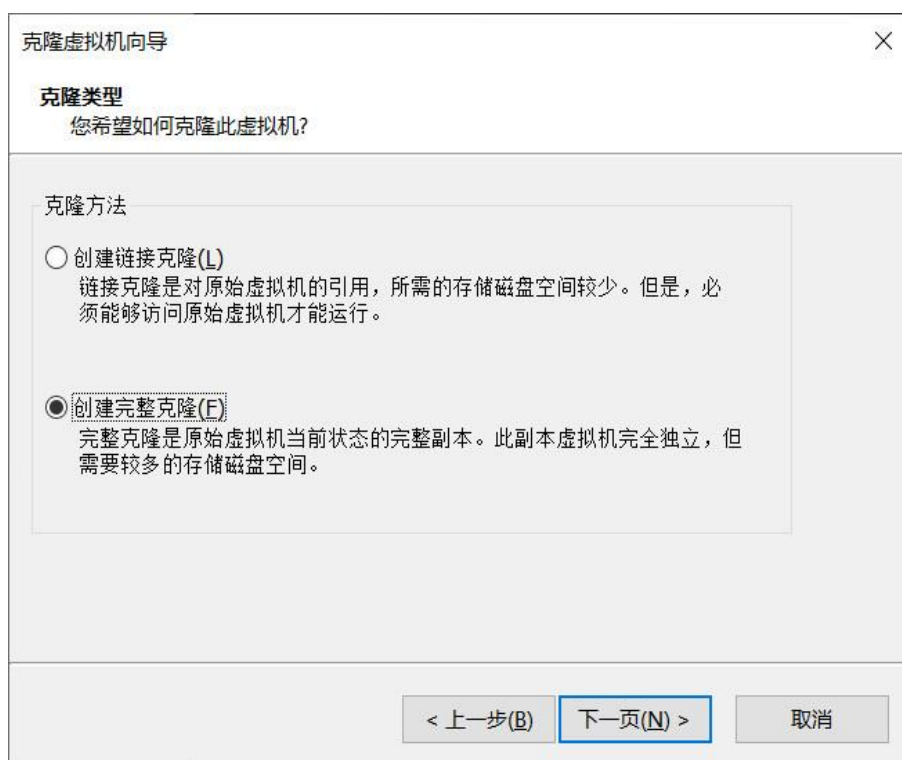


图 2-65 克隆虚拟机方法

步骤 04: 在新建虚拟机名称界面, 输入虚拟机名称为“slave1”, 然后单击【浏览 (V)】为即将创建的虚拟机 slave1 选择存储路径, 然后单击【完成】按钮, 如图 2-66 所示, 接下来就开始自动克隆虚拟机, 最后在虚拟机克隆完成后, 单击【关闭】按钮即可, 如图 2-67 所示。

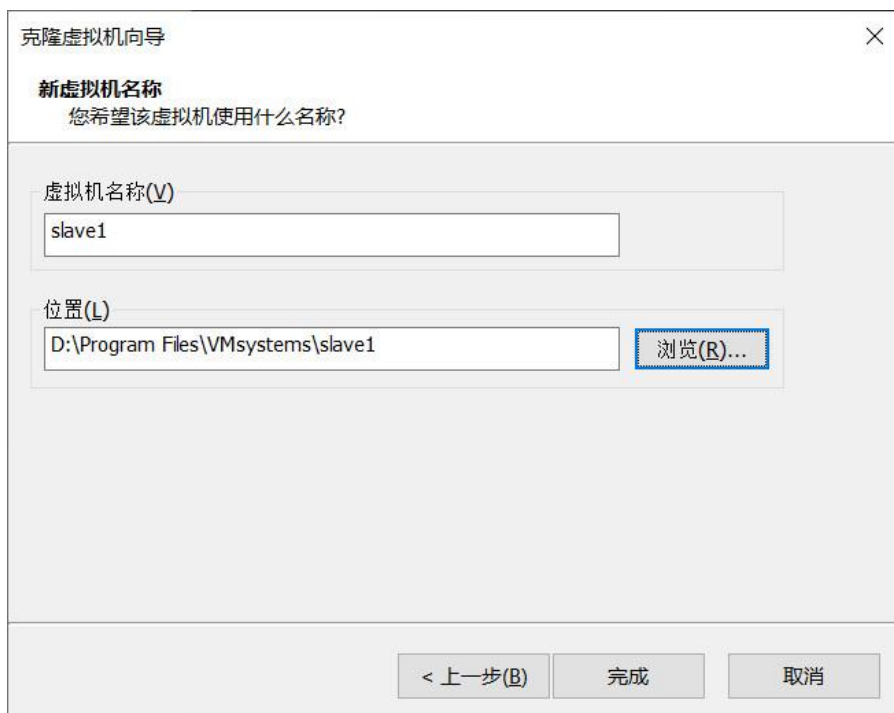


图 2-65 新虚拟机名称

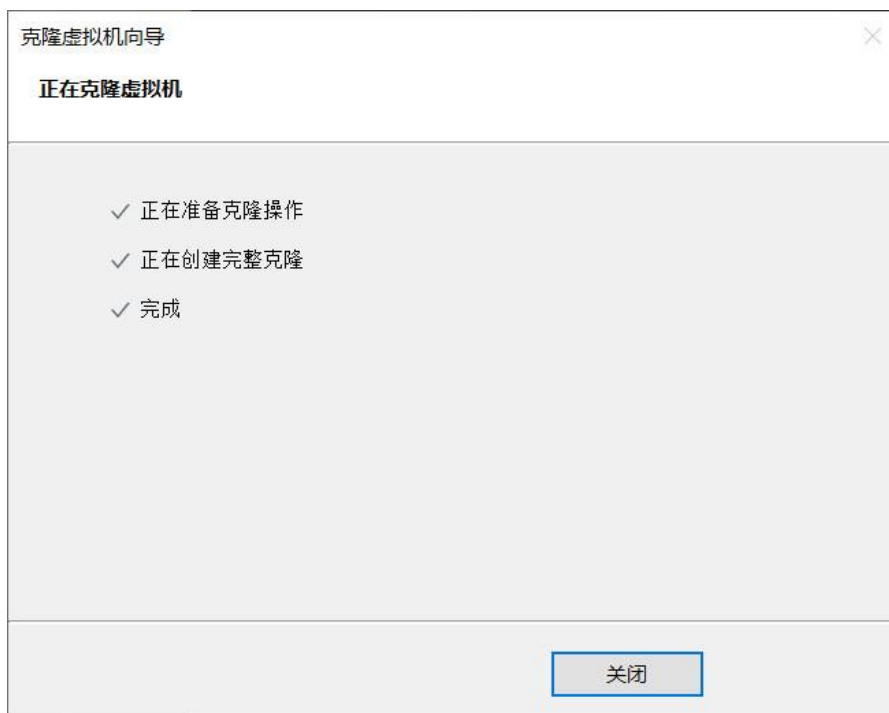


图 2-67 虚拟机克隆完成

2.4.3 修改配置文件

在开启虚拟机 slave1 之后，接下来需要为虚拟机 slave1 进行一些必要的配置工作，如 ip 地址、主机名。

步骤 01：修改主机名，修改方法详见 1.2.2 小节中步骤 17，修改后如图 2-68 所示。



图 2-68 修改主机名

步骤 02: 在终端中执行命令 “rm -rf /etc/udev/rules.d/70-persistent-ipoib.rules”，删除 70-persistent-ipoib.rules 文件。

步骤 03: 修改/etc/sysconfig/network-scripts/ifcfg-ens33 文件，为虚拟机 slave1 分配新的 ip 地址，修改后的内容如下所示。

```
TYPE=Ethernet
BROWSER_ONLY=no
DEFROUTE=yes
NAME=ens33
DEVICE=ens33
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.75.101    # 为 slave1 分配 192.168.75.101 地址
NETMASK=255.255.255.0
GATEWAY=192.168.75.2
DNS1=114.114.114.114
DNS2=8.8.8.8
```

步骤 04: 修改/etc/sysconfig/network 文件，修改虚拟机的主机名，修改后的内容如下所示。

```
# Created by anaconda
NETWORKING=yes
HOSTNAME=slave1
```

步骤 05: 完成上述修改后，在终端中执行 “reboot” 重启虚拟机。

步骤 06: 验证配置是否正确无误，在终端中分别执行 “ping master”、“ping www.baidu.com”，如图 2-69、图 2-70 所示。

```
[root@slave1 ~]# ping master
PING master (192.168.75.100) 56(84) bytes of data.
64 bytes from master (192.168.75.100): icmp_seq=1 ttl=64 time=0.559 ms
64 bytes from master (192.168.75.100): icmp_seq=2 ttl=64 time=0.190 ms
64 bytes from master (192.168.75.100): icmp_seq=3 ttl=64 time=0.203 ms
64 bytes from master (192.168.75.100): icmp_seq=4 ttl=64 time=0.349 ms
```

图 2-69 ping master 测试

```
[root@slave1 ~]# ping www.baidu.com
PING www.a.shifen.com (39.156.66.14) 56(84) bytes of data.
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=1 ttl=128 time=18.8 ms
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=2 ttl=128 time=17.2 ms
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=3 ttl=128 time=17.6 ms
64 bytes from 39.156.66.14 (39.156.66.14): icmp_seq=4 ttl=128 time=20.5 ms
```

图 2-70 ping 网络测试

(7) 重复 (1) ~ (5) 的步骤，克隆虚拟机 slave2，并进行相关配置修改和验证。

2.4.4 配置 SSH 免密登录

SSH 为 Secure Shell 的缩写，SSH 为建立在应用层基础上的安全协议，专为远程登录会话和其他网络服务提供安全性的协议。SSH 客户端适用于多种平台，几乎所有 UNIX

平台—包括 HP-UX、Linux、AIX、Solaris，以及其他平台，都可运行 SSH。配置 SSH 可以集群节点之间的免密登录，配置步骤如下，以下配置步骤均在 master 节点执行。

步骤 01：生成公钥和私钥对

在终端中输入命令“ssh-keygen -t rsa”，然后按三次“Enter”键，生成私有密钥 id_rsa 和公钥密钥 id_rsa.pub 文件，如图 2-71 所示。

```
[root@master ~]# 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: XtYaXceorp8zwd9ntW0warxyFxpjn9Gjfn+GL4dXiw root@master
The key's randomart image is:
+---[RSA 2048]-----+
|
|                o
|               + o
|              o o +
|             S = + o +
|            . + . = + +
|           o + .XoE=
|          . . . * . + +
|         +o .o . * +
+---[SHA256]-----+
```

图 2-71 生成生成私有密钥和公钥密钥

步骤 02：将公钥复制到其他机器中

使用 ssh-copy-id 命令，将 master 生成的公钥复制到其他虚拟机中，代码如下所示，执行结果如图 2-72 所示。

```
ssh-copy-id -i /root/.ssh/id_rsa.pub master
ssh-copy-id -i /root/.ssh/id_rsa.pub slave1
ssh-copy-id -i /root/.ssh/id_rsa.pub slave2
```

```
[root@master ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'slave2 (192.168.75.102)' can't be established.
ECDSA key fingerprint is SHA256: N1y3LJgKV3Pg11FDwB8CF1+g9Vl3+YuBEalaUibbewg.
ECDSA key fingerprint is MD5: b7:29:46:d4:76:94:4b:29:a0:5d:29:67:f4:93:1c:38.
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@master ~]#
```

图 2-72 修改 ifcfg-ens33

步骤 03：验证 SSH 是否配置成功

在 master 终端中，分别使用 ssh 去登录 slave1 和 slave2，如图 2-73 所示，如果能够免密登录，说明配置成功。

```

[root@master ~]# ssh slavel
Last login: Mon Aug 9 22:37:40 2021 from master
[root@slavel ~]# exit
登出
Connection to slavel closed.
[root@master ~]# ssh slave2
Last login: Mon Aug 9 22:12:26 2021
[root@slave2 ~]# exit
登出
Connection to slave2 closed.
[root@master ~]#

```

图 2-73 ssh 切换测试

2.4.5 HDFS 初始格式化

完成所有 Hadoop 相关配置之后，在首次启动 Hadoop 集群之前需要对 NameNode 执行格式化操作，该操作会初始化 HDFS 相关的配置。

在 master 终端中执行如下所示代码来格式化 NameNode。

```

cd $HADOOP_HOME/sbin
hadoop namenode -format

```

执行完格式化命令后，若在最后出现 “Storage directory /data/hadoop/hdfs/name has been successfully formatted” 提示，则格式化成功，如图 2-74 所示。

```

21/08/09 22:49:09 INFO util.GSet: capacity = 2^15 = 32768 entries
21/08/09 22:49:09 INFO namenode.NNConf: ACLs enabled? false
21/08/09 22:49:09 INFO namenode.NNConf: XAttrs enabled? true
21/08/09 22:49:09 INFO namenode.NNConf: Maximum size of an xattr: 16384
21/08/09 22:49:10 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1378352390-192.168.75.100-1628520549966
21/08/09 22:49:10 INFO common.Storage: Storage directory /usr/local/src/hadoop-2.6.5/dfs/name has been successfully formatted.
21/08/09 22:49:10 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/src/hadoop-2.6.5/dfs/name/current/fsimage.ckpt_000000000000000000 using no compression
21/08/09 22:49:10 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/src/hadoop-2.6.5/dfs/name/current/fsimage.ckpt_000000000000000000 of size 321 bytes saved in 0 seconds.
21/08/09 22:49:10 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
21/08/09 22:49:10 INFO util.ExitUtil: Exiting with status 0
21/08/09 22:49:10 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.75.100

```

图 2-74 格式化 HDFS

2.4.6 启动 Hadoop

成功格式化完成之后，就可以启动 Hadoop 集群，启动集群同样只需要在 master 中完成，在 master 终端中输入如下代码。

```

cd $HADOOP_HOME/sbin
./start-all.sh # start-all.sh 依次执行 start-dfs.sh、start-yarn.sh

```

启动集群如图 2-75 所示。

```
[root@master sbin]# ./start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/src/hadoop-2.6.5/logs/hadoop-root-namenode-master.out
slave1: starting datanode, logging to /usr/local/src/hadoop-2.6.5/logs/hadoop-root-datanode-slave1.out
slave2: starting datanode, logging to /usr/local/src/hadoop-2.6.5/logs/hadoop-root-datanode-slave2.out
Starting secondary namenodes [master]
master: starting secondarynamenode, logging to /usr/local/src/hadoop-2.6.5/logs/hadoop-root-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/src/hadoop-2.6.5/logs/yarn-root-resourcemanager-master.out
slave2: starting nodemanager, logging to /usr/local/src/hadoop-2.6.5/logs/yarn-root-nodemanager-slave2.out
slave1: starting nodemanager, logging to /usr/local/src/hadoop-2.6.5/logs/yarn-root-nodemanager-slave1.out
[root@master sbin]#
```

完成集群启动之后分别在 master、slave1、slave2 节点执行命令“jps”查看 Hadoop 进程，如图 2-76 所示。

```
root@slave2:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master ~]# jps
5362 ResourceManager
5052 NameNode
5228 SecondaryNameNode
8974 Jps
[root@master ~]# ssh slave1
Last login: Sat Aug 14 17:32:52 2021 from master
[root@slave1 ~]# jps
6101 Jps
4569 NodeManager
4479 DataNode
[root@slave1 ~]# ssh slave2
Last login: Mon Aug 9 22:42:17 2021 from slave2
[root@slave2 ~]# jps
4197 DataNode
5164 Jps
4287 NodeManager
[root@slave2 ~]#
```

图 2-76 查看节点进程

温馨提示：

启动 Hadoop 集群顺序：

```
cd $HADOOP_HOME/sbin
start-dfs.sh
start-yarn.sh
```

关闭 Hadoop 集群顺序：

2.4.7 Hadoop 集群监控

Hadoop 集群常见相关服务监控默认端口及地址如表 2-3 所示。

表 2-3 Hadoop 服务端口地址

| 服务名称 | 默认端口 | Web 地址 |
|------------------|-------|----------------------------------|
| NameNode | 50070 | http://namenode_ip:port/ |
| ResourceManager | 8088 | http://resourcemanager_ip:port/ |
| JobHistoryServer | 19888 | http://jobhistoryserver_ip:port/ |

温馨提示：

如果需要在 Windows 的浏览器中打开 Hadoop 监控页面可以使用“节点 ip 地址:对应服务端口”格式地址，若需要使用虚拟机主机名，则同理需要在

Windows 中配置 hosts 文件，本地 hosts 文件在

(1) HDFS 监控

在 Hadoop 集群正常启动之后，通过 HDFS UI 监控界面可以方便的进行集群存储的查看，它默认开启的端口为 50070。

在浏览器中的地址栏中输入 http://192.168.75.100:50070（或 http://master:50070），按回车即可进入到 HDFS 监控页面，如图 2-77 所示。

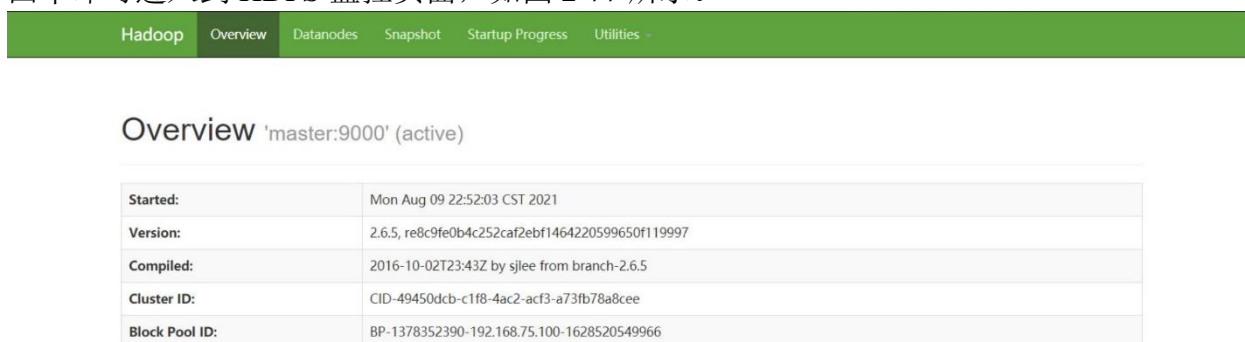


图 2-77 hadoop web UI

这俩菜单栏主要包括的内容分别是：Overview（集群概述）、Datanodes（数据节点）、Datanode Volume Failures（数据节点卷故障）、Snapshot（快照）、Snapshot Progress（启动进度）、Utilities（工具页）。而 HDFS 文件存储目录及内容在 Utilities 菜单下。

在 HDFS 监控页面顶部菜单栏中，选择【Utilities】—>【Browse the file system】菜单，可以进入到 HDFS 目录，查看 HDFS 的目录结构和存储的文件，如图 2-78 所示。

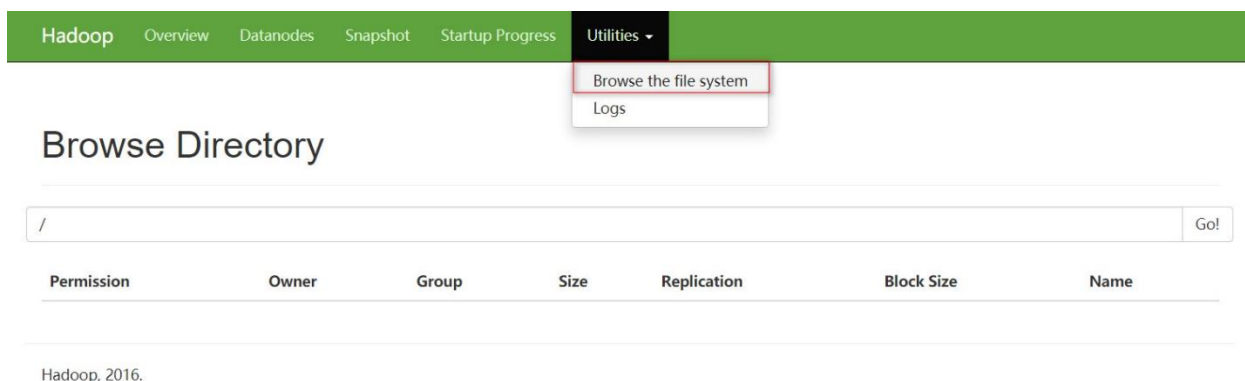


图 2-78 查看存储目录

(2) Yarn 监控

在浏览器中的地址栏中输入 `http://192.168.75.100:8088` (或 `http://master:8088`)，按回车即可进入到 Yarn 监控页面，如图 2-79 所示。Yarm 监控页面展示了集群的任务详情，包括当前运行任务、已完成任务等。

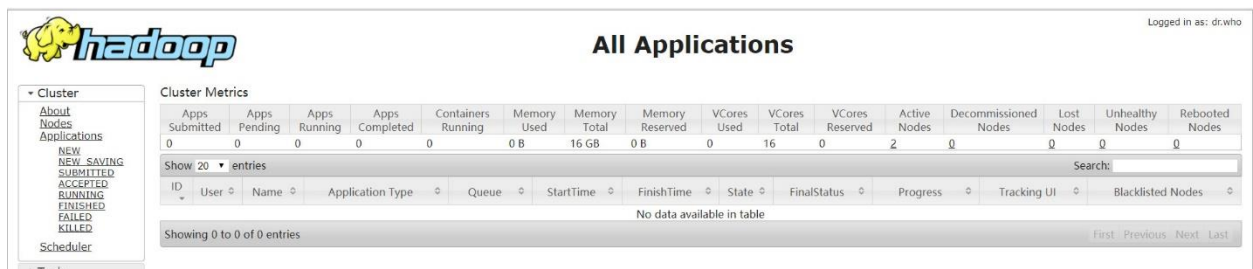


图 2-79 查看 Yarn 监控

★回顾思考★

01 vim 编辑器常用的命令有哪些？

答：包括如下：

- (1) 按“i”进入编辑状态；
- (2) 按“Esc”退出编辑状态；
- (3) 按“:q”退出不保存；
- (4) 按“:q!”强制退出不保存；
- (5) 按“:wq”保存并退出；

02 Hadoop 完全分布式集群都需要启动哪些服务？

答：master 节点：NameNode、SecondaryNameNode、ResourceManager、JobHistoryServer(执行“`mr-jobhistory-daemon.sh start historyserver`”)。slave 节点：DataNode、NodeManager。

★练习★

一、选择题

1. Hadoop 集群启动的先后顺序为 ()

- ① `start-all.sh`
 - ② `start-yarn.sh`
 - ③ `start-dfs.sh`
- A. ①②③
B. ③②①
C. ③①

- D. ③②
2. Hadoop 集群关闭的先后顺序为 ()
- ① stop-all.sh
 - ② stop-yarn.sh
 - ③ stop-dfs.sh
- A. ①②③
- B. ③②①
- C. ③①
- D. ③②
3. HDFS 配置中默认备份数为 () 份
- A. 1
- B. 2
- C. 3
- D. 4
4. Yarn 监控的默认端口是 ()
- A. 8088
- B. 8000
- C. 19888
- D. 50070
5. mapred-site.xml 文件的作用是 ()
- A. 配置 Yarn 框架
- B. 配置 MapReduce 框架
- C. 配置 HDFS 系统的相关内容
- D. 配置子节点信息

二、填空题

1. vim 修改并保存命令是_____。
2. Hadoop 配置文件中配置 Yarn 框架的文件是_____。

三、实战练习

1. 使用 vim 编辑器在 master 中 “/usr/local/” 目录下编写文件 demo.txt 并保存。
2. 搭建 Hadoop 完全分布式集群。

本章小结

本章详细介绍了虚拟机的安装配置、Java 的安装以及 Hadoop 完全分布式集群的搭建。首先，在 Linux 中 JDK、Hadoop 安装过程，简要介绍了 vim 和 gedit 文本编辑工具的使用，熟悉 Linux 基本操作是学习大数据的必备技能。其次，介绍了在虚拟机安装过程中对虚拟机固定 IP 设置的方法。最后，在搭建 Hadoop 完全分布式环境的小节中完整详细

的介绍了 Hadoop 的安装、配置、格式化、启动过程，这部分也是本章的重点，掌握 Hadoop 完全分布式的安装，也是学习大数据的基础。