南京信息工程大学　实验（实习）报告

实验（实习）名称　大数据课程设计　（实习）日期　6-9　得分＿＿＿＿＿＿

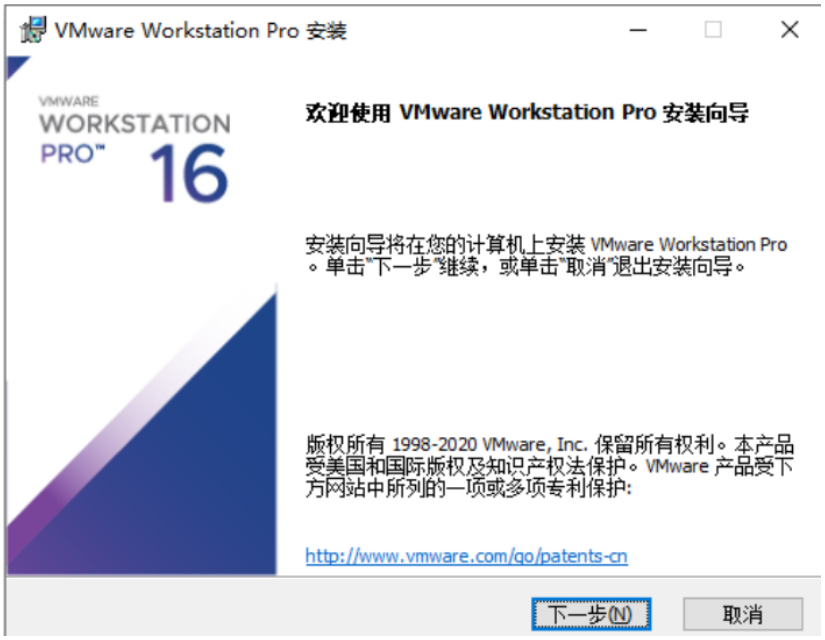
指导教师　孙乐　专业　软件工程（中外合作办学）　年级　2020 级　班次　2

姓名　颜晓雨　学号　202083020070

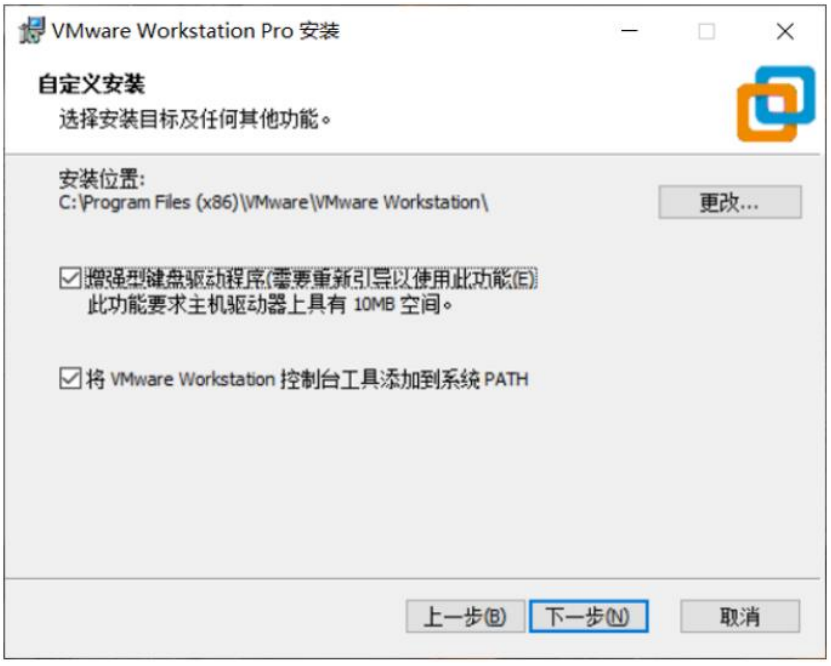
实验内容及步骤：

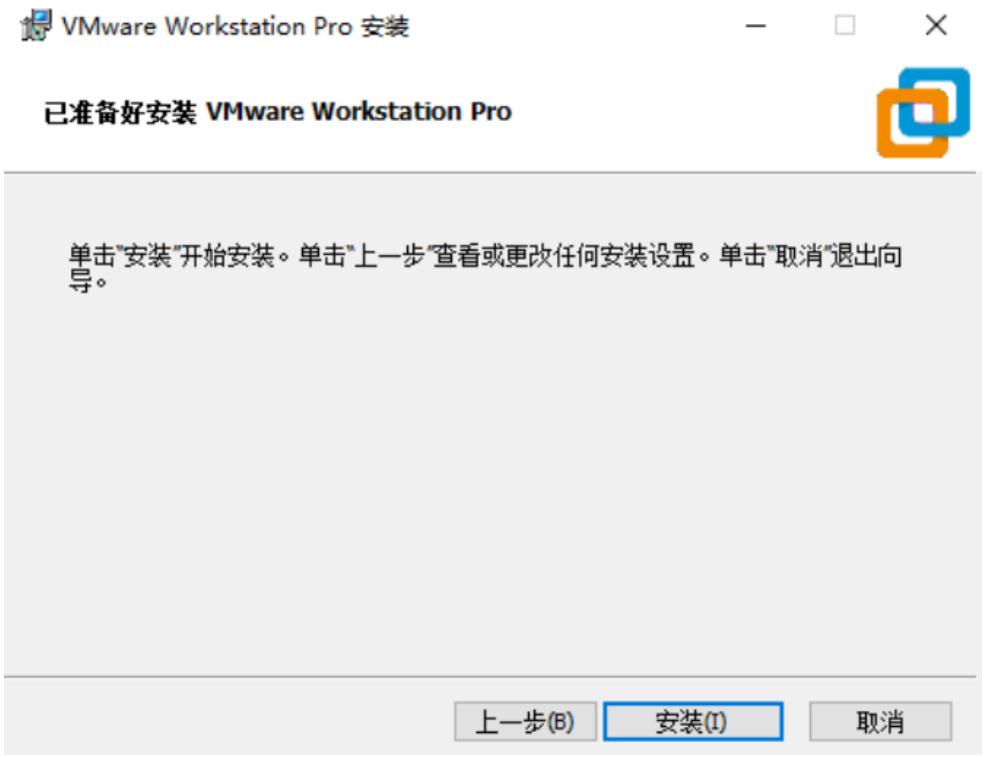
1.系统环境和软件安装：包括 Linux 操作系统的安装过程、所选择的 Linux 发行版、Hadoop 分布式系统的安装过程以及其他必要组件的安装。

安装 VM 虚拟机：

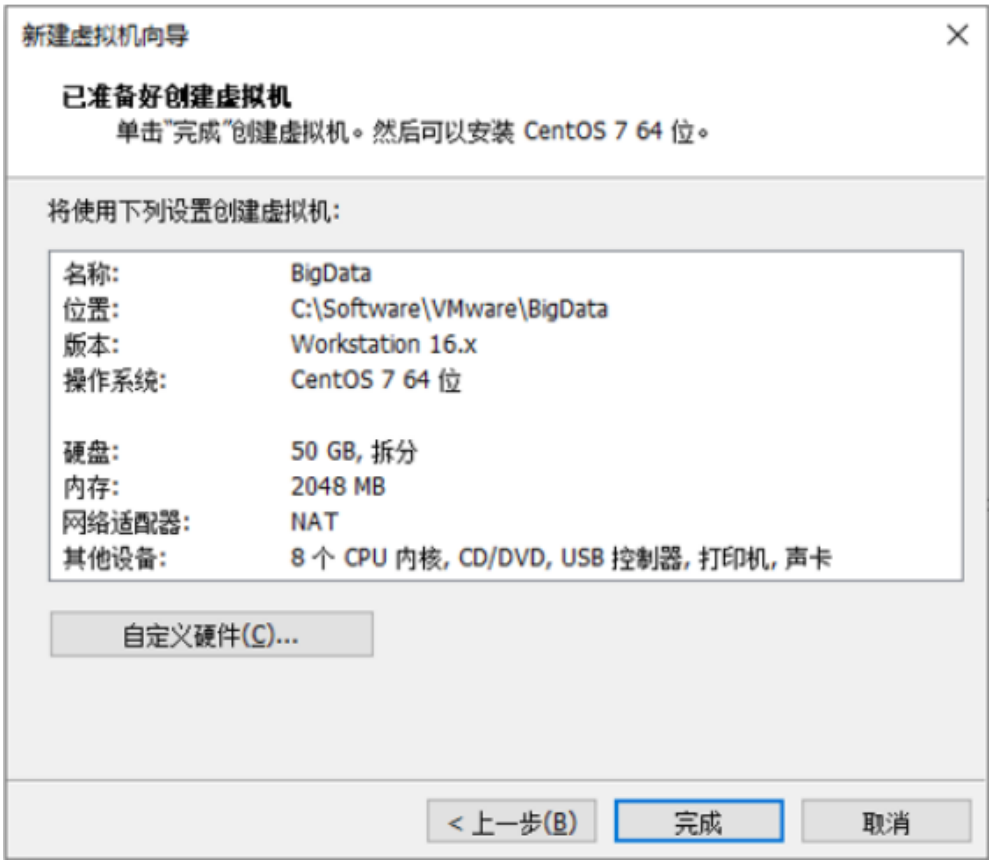


安装步骤 1



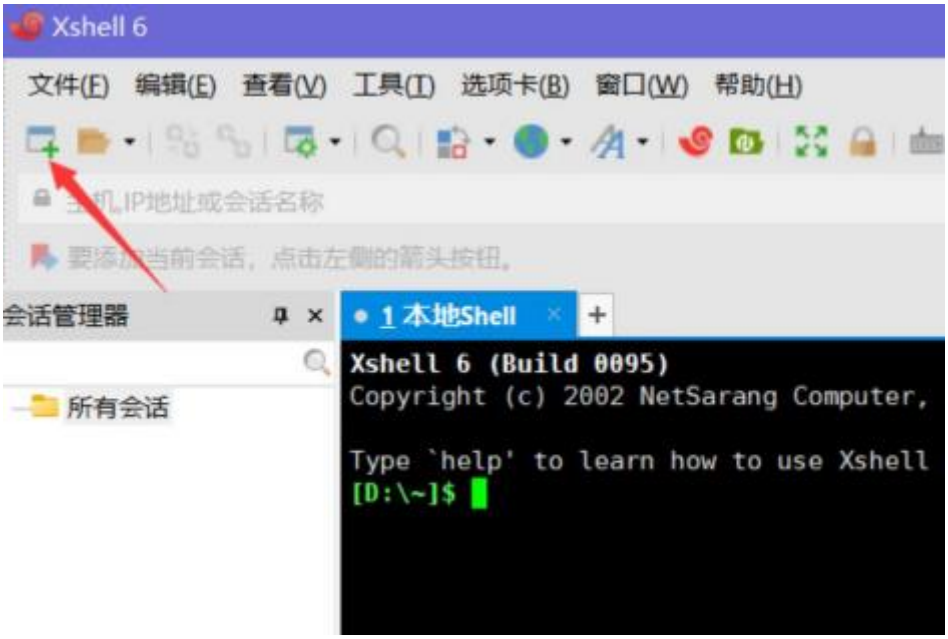

VMware Workstation Pro 安装
已准备好安装 VMware Workstation Pro
单击"安装"开始安装。单击"上一步"查看或更改任何安装设置。单击"取消"退出向导。
上一步(B)
安装(I)
取消

安装步骤 2


新建虚拟机向导
已准备好创建虚拟机
单击"完成"创建虚拟机。然后可以安装 CentOS 7 64 位。
将使用下列设置创建虚拟机:
名称: BigData
位置: C:\Software\VMware\BigData
版本: Workstation 16.x
操作系统: CentOS 7 64 位
硬盘: 50 GB, 拆分
内存: 2048 MB
网络适配器: NAT
其他设备: 8 个 CPU 内核, CD/DVD, USB 控制器, 打印机, 声卡
自定义硬件(C)...
< 上一步(B)
完成
取消

安装完成

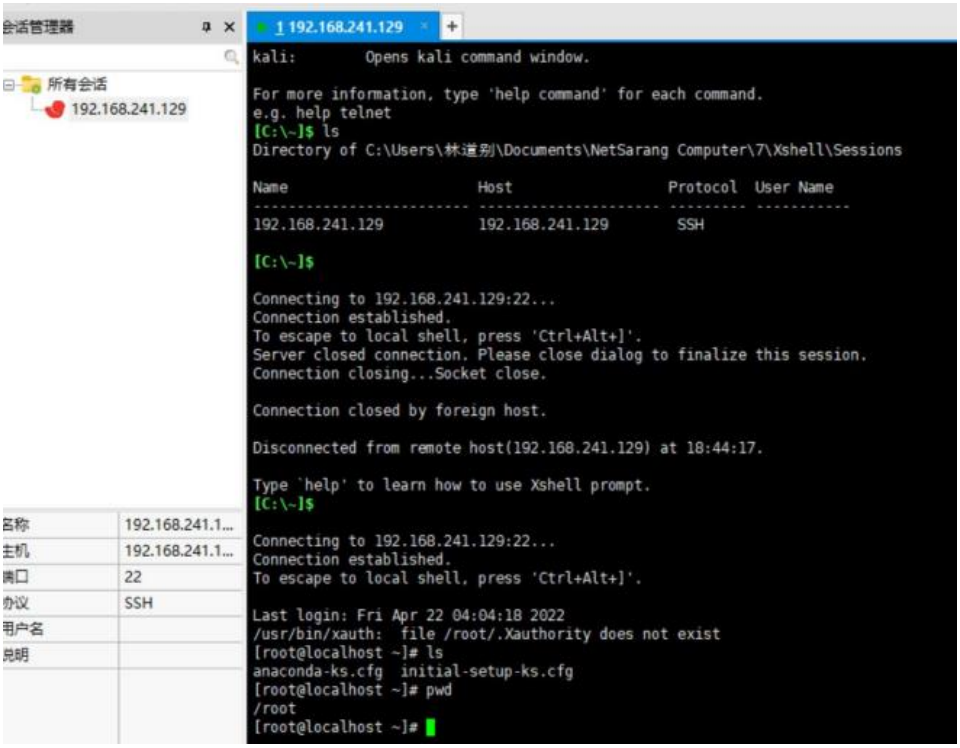
Xshell 访问 Linux：



新建

SSH 钥匙准备链接

输入账号密码



远程服务链接成功

安装 centos:

新建虚拟机

完成新建

2.Hadoop 集群配置：说明搭建的 Hadoop 集群的架构、节点角色和配置文件的设置。应包括主节点（NameNode）和从节点（DataNode）的配置，以及资源管理器（ResourceManager）和节点管理器（NodeManager）的配置。

说明：

搭建 Hadoop 集群所需的节点包括 NameNode、Secondary NameNode、DataNode、ResourceManager 和 NodeManager。其中 NameNode 和 Secondary NameNode 运行在主节点，DataNode 和 NodeManager 则运行在从节点。

主节点架构包含 NameNode、ResourceManager 和 Secondary NameNode，从节点架构包含 DataNode 和 NodeManager。

以下是每个节点的角色和配置文件设置：

1）NameNode

角色：管理整个分布式文件系统的元数据。

配置文件设置：

在 hdfs-site.xml 配置文件中设置以下参数：

- dfs.namenode.name.dir：指定 NameNode 的元数据存储目录。
- dfs.replication：指定文件块的副本数量。
- dfs.permissions.enabled：启用 HDFS 的权限管理功能。
- dfs.block.size：指定文件块的大小。

在 core-site.xml 配置文件中设置以下参数：

- fs.defaultFS：指定 Hadoop 集群使用的默认文件系统 URI。
- hadoop.tmp.dir：指定临时目录。

2）Secondary NameNode

角色：定期将 NameNode 的元数据切换到新的编辑日志文件。

配置文件设置：

在 hdfs-site.xml 配置文件中设置以下参数：

- dfs.namenode.secondary.http-address：指定 Secondary NameNode 的 Web 管理界面的地址。
- dfs.namenode.checkpoint.dir：指定 Secondary NameNode 的元数据存储目录。
- dfs.namenode.checkpoint.period：指定进行元数据检查点的时间间隔。

3）DataNode

角色：储存文件块的实际数据。

配置文件设置：

在 hdfs-site.xml 配置文件中设置以下参数：

- dfs.datanode.data.dir：指定 DataNode 的数据存储目录

在 core-site.xml 配置文件中设置以下参数：

- hadoop.tmp.dir：指定临时目录。

4）ResourceManager

角色：管理 Hadoop 集群上的资源和应用程序。

配置文件设置：

在 yarn-site.xml 配置文件中设置以下参数：

- yarn.nodemanager.aux-services：指定 NodeManager 服务所需的辅助服务。
- yarn.nodemanager.aux-services.mapreduce.shuffle.class：指定 MapReduce Shuffle 插件的实现类。
- yarn.resourcemanager.address：指定 ResourceManager 的地址。
- yarn.resourcemanager.scheduler.address：指定资源分配的调度程序的地址。
- yarn.resourcemanager.resource-tracker.address：指定 ResourceManager 的资源跟踪器地址。
- yarn.resourcemanager.admin.address：指定 ResourceManager 的管理地址。
- yarn.nodemanager.local-dirs：指定本地临时目录。
- yarn.nodemanager.log-dirs：指定日志文件存储目录。

5）NodeManager

角色：管理从节点上的资源和应用程序。

配置文件设置：

在 yarn-site.xml 配置文件中设置以下参数：

- yarn.nodemanager.local-dirs：指定本地临时目录。
- yarn.nodemanager.log-dirs：指定日志文件存储目录。

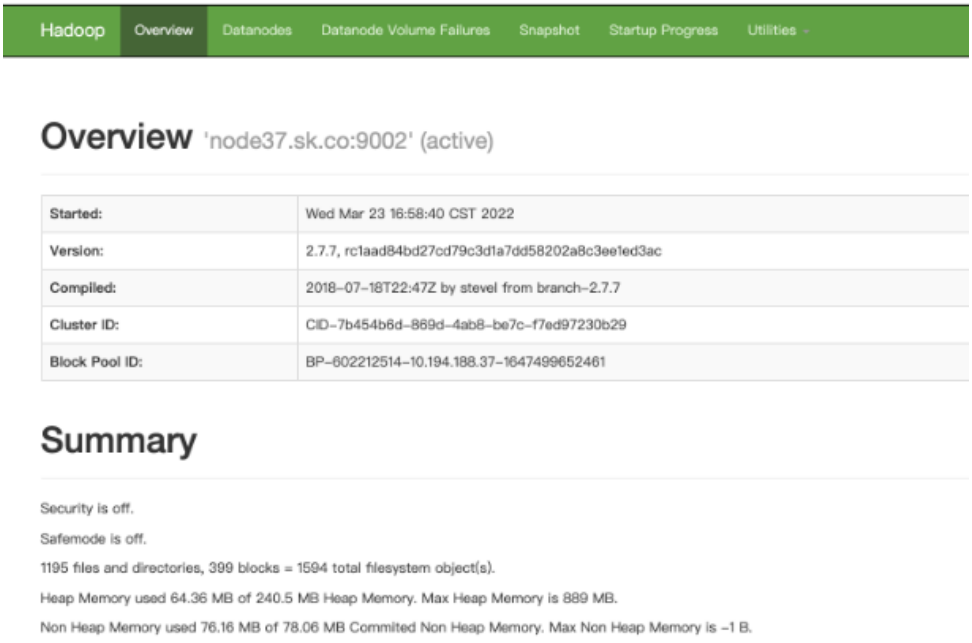
安装 Hadoop 软件：

```
tar -zxf hadoop-2.6.0-cdh5.14.2.tar.gz
mv hadoop-2.6.0-cdh5.14.2 soft/hadoop260
```

```
[root@hx opt]# cd soft
[root@hx soft]# ls
elasticsearch622  elasticsearchhead  hadoop260  jdk180  logstash622  maven361  nodev11  tomcat85
```

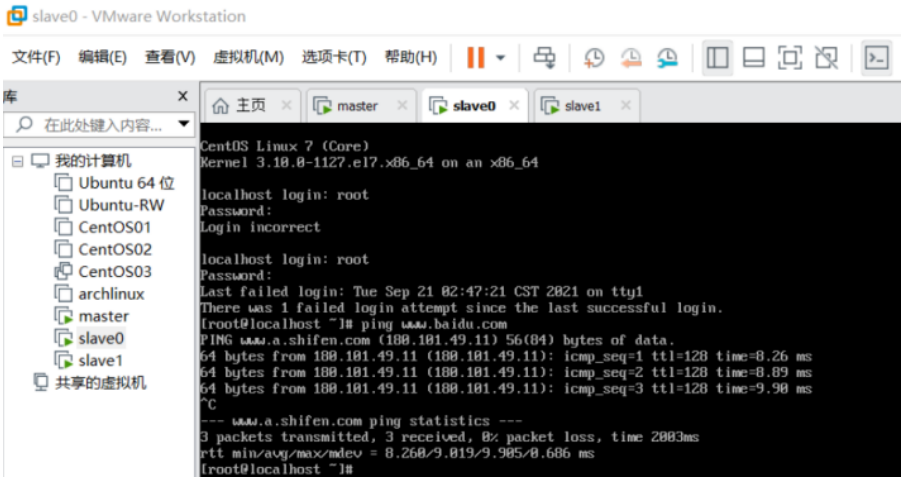
解压安装包至软件目录

start-all.sh 开启 hadoop 集群，stop-all.sh 关闭集群

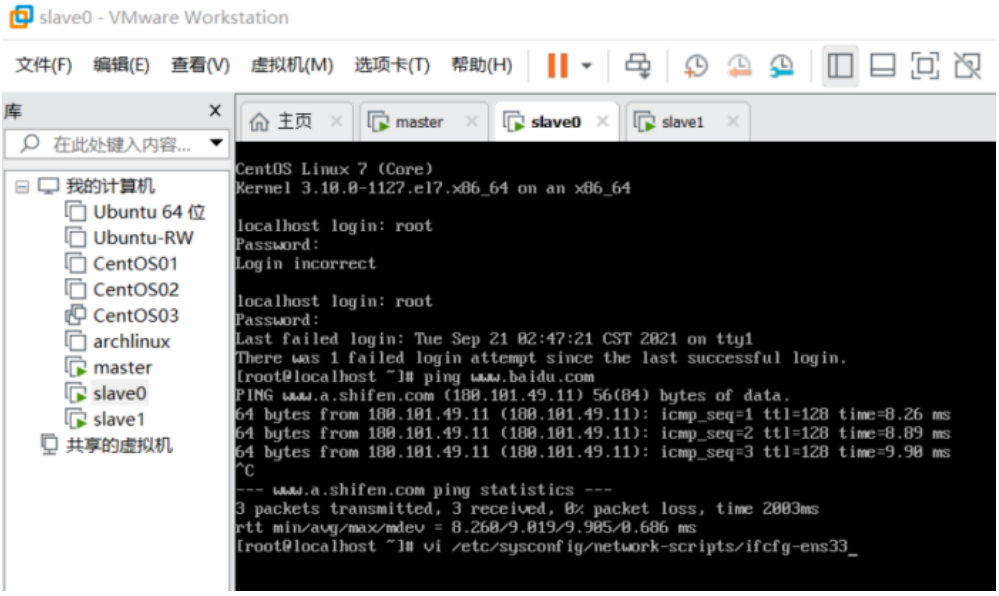


搭建完成

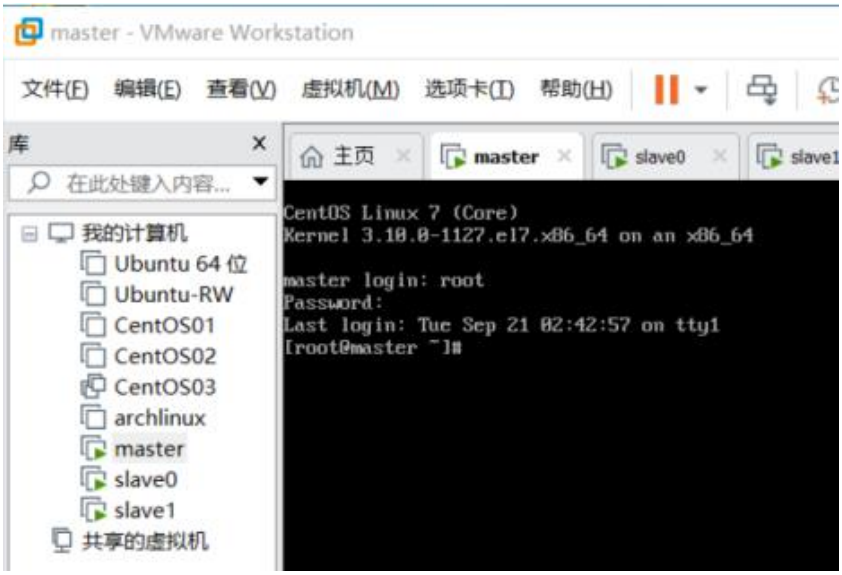
配置网络连接：



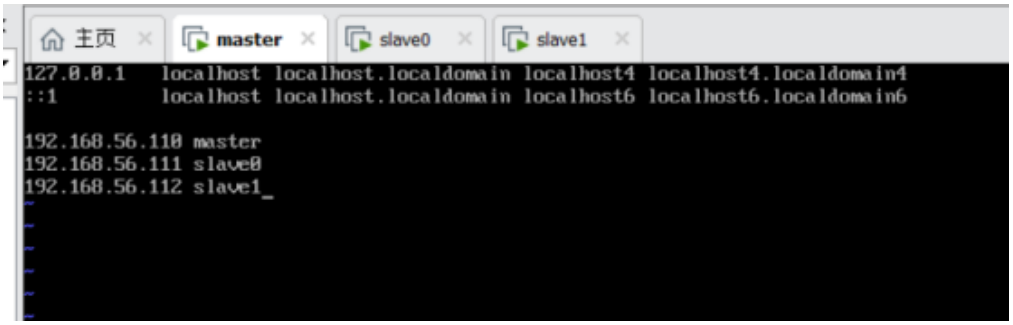
配置静态网络，关闭防火墙



完成



配置主机名和 hosts 文件



完成

bin目录：存放对Hadoop相关服务（HDFS,YARN）进行操作的脚本

etc目录：Hadoop的配置文件目录，存放Hadoop的配置文件

lib目录：存放Hadoop的本地库（对数据进行压缩解压缩功能）

sbin目录：存放启动或停止Hadoop相关服务的脚本

share目录：存放Hadoop的依赖jar包、文档、和官方案例

重要目录

3.集群测试和操作：使用 wordcount 示例程序对 Hadoop 集群进行测试，展示运行结果并解释过程。同时，展示 Hadoop 集群中关键组件 HDFS、YARN 和 HBase 的 Web UI 界面截图，说明其功能和使用方法。

Hadoop 集群启动：

```
[root@hadoop ~]# hdfs namenode -format
```

```
[root@hadoop ~]# jps
```

启动 Hadoop

```
[root@hadoop ~]# start-dfs.sh
```

启动 HDFS

在 Hadoop 集群上运行 WordCount 程序：

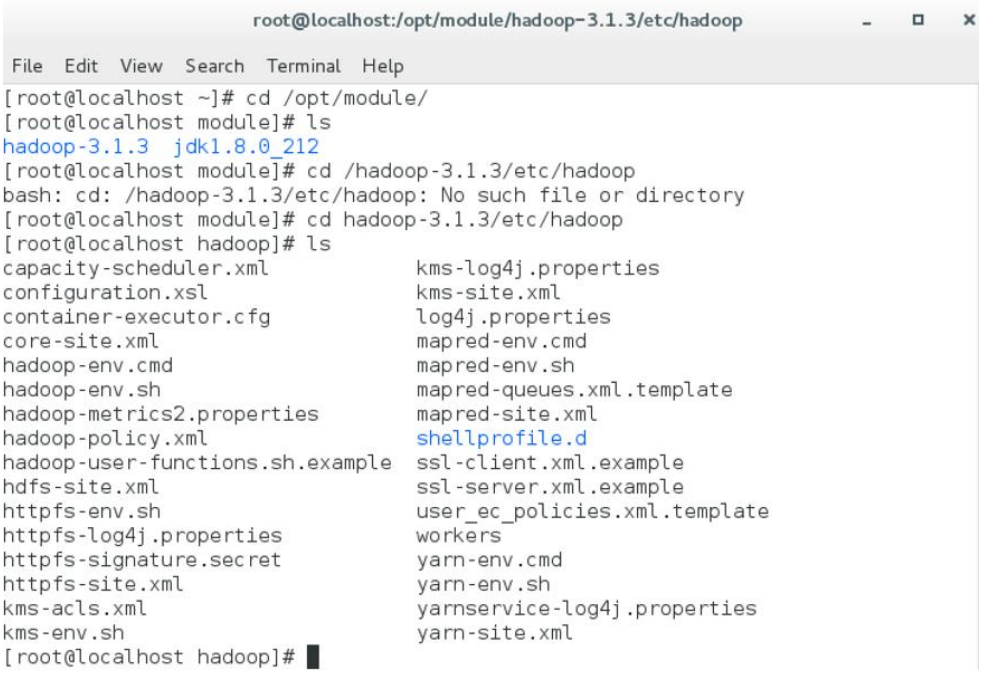
```
$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount input output
```

从 HDFS 获取输出文件：

```
$ hadoop fs -get output
```

运行结果：

WordCount 程序对输入文件进行了单词计数，输出结果包括单词和单词出现次数。结果存储在 HDFS 的/output 目录下。



环境路径

HDFS Web UI：

HDFS Web UI 是 HDFS 的 Web 管理界面，可通过浏览器远程访问。通过该界面，用户可以监控和管理 HDFS 集群的运行状况。

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
#0.0.0.0 account.jetbrains.com
#0.0.0.0 www.jetbrains.com
```

可以看到 HDFS Web UI 提供了很多有用的信息，如活跃的和故障的节点、存储池使用情况、文件系统状态等。用户可以从中了解 HDFS 集群的情况，从而进行调整和优化。

YARN Web UI：

YARN Web UI 是 YARN 的 Web 管理界面，可通过浏览器远程访问。通过该界面，用户可以监控和管理 YARN 集群的运行状况。

Open ▾ | yarn-site.xml | /opt/module/hadoop-3.1.3/etc/hadoop | Save

```
  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.
-->
<configuration>
<!-- resourcemanager的主机名 -->
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>hadoop</value>
    </property>
<!-- 分配给容器的物理内存量，单位是MB，设置为-1则自动分配，默认8192MB -->
    <property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>4096</value>
    </property>
<!-- NodeManager上运行的服务列表,可以配置成mapreduce_shuffle，多个服务使用逗号隔开 -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
    <property>
        <name>yarn.application.classpath</name>
        <value>/opt/module/hadoop-3.1.3/etc/hadoop:/opt/module/hadoop-3.1.3/share/hadoop/
common/lib/*:/opt/module/hadoop-3.1.3/share/hadoop/common/*:/opt/module/hadoop-3.1.3/share/
hadoop/hdfs:/opt/module/hadoop-3.1.3/share/hadoop/hdfs/lib/*:/opt/module/hadoop-3.1.3/share/
hadoop/hdfs/*:/opt/module/hadoop-3.1.3/share/hadoop/mapreduce/lib/*:/opt/module/hadoop-3.1.3/
share/hadoop/mapreduce/*:/opt/module/hadoop-3.1.3/share/hadoop/yarn:/opt/module/hadoop-3.1.3/
share/hadoop/yarn/lib/*:/opt/module/hadoop-3.1.3/share/hadoop/yarn/* </value>
    </property>
</configuration>
```

可以看到 YARN Web UI 提供了很多有用的信息，如集群节点、正在运行的作业、队列情况等。用户可以从中了解 YARN 集群的情况，从而进行调整和优化。

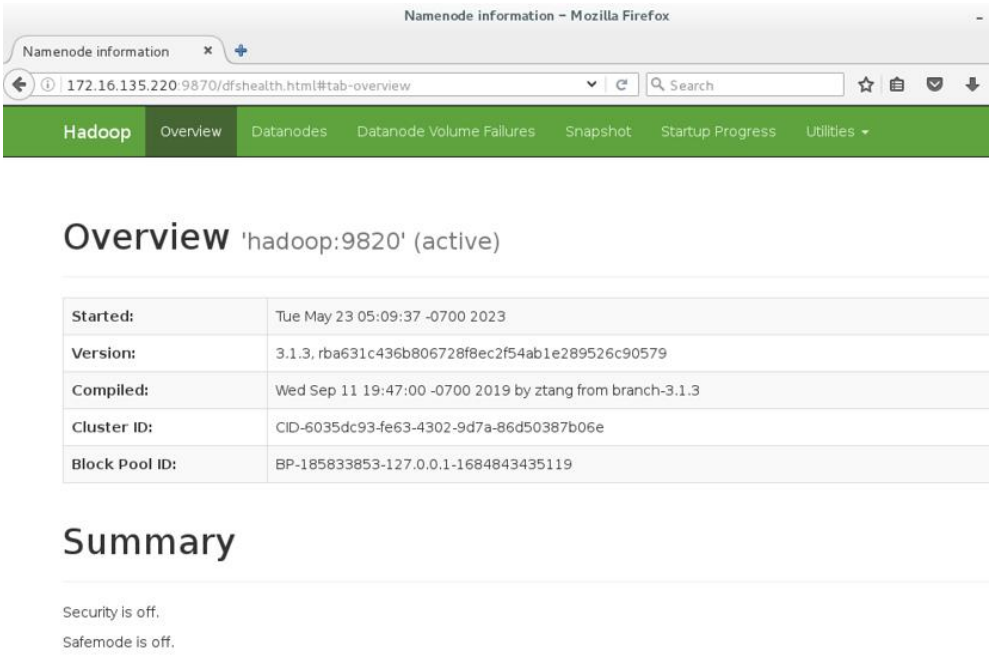
HBase Web UI：

HBase Web UI 是 HBase 的 Web 管理界面，可通过浏览器远程访问。通过该界面，用户可以监控和管理 HBase 集群的运行状况。

Open ▾ | hdfs-site.xml | /opt/module/hadoop-3.1.3/etc/hadoop | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<?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>
        <property>
        <name>dfs.namenode.http-address</name>
        <value>hadoop:9870</value>
        </property>
        <property>
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
        </property>
</configuration>
```

可以看到 HBase Web UI 提供了很多有用的信息，如表名、表结构、region 分布等。用户可以从中了解 HBase 集群的情况，从而进行调整和优化。



访问 UI 界面

执行作业，测试 wordcount:

关闭 Hadoop 集群:

```
[root@hadoop ~]# stop-dfs.sh
```

```
[root@localhost hadoop]# stop-dfs.sh
Stopping namenodes on [hadoop]
Last login: Tue May 23 05:09:55 PDT 2023 on pts/0
Stopping datanodes
Last login: Tue May 23 05:23:11 PDT 2023 on pts/1
```

```
[root@hadoop ~]# stop-yarn.sh
```

4.问题回答：对以下问题进行回答，并给出详细的解答。

1）分布式的理解：对分布式系统的概念和原理进行解释，阐述其优势和应用领域。

概念：分布式系统是由多个独立计算机组成的系统，这些计算机通过网络互相通信和协作，共同完成一个任务。分布式系统的设计目标是提高系统的可靠性、可扩展性和性能。

分布式系统的原理包括：

a.通信原理：分布式系统中的计算机通过网络通信来协作完成任务。

b.协调原理：分布式系统中的计算机需要协调彼此的行为，以达到共同的目标。

c.一致性原理：分布式系统中的计算机需要保持数据的一致性，以确保系统的正确性。

d.容错原理：分布式系统中的计算机需要具备容错能力，以保证系统的可靠性。

分布式系统的优势包括：

a.可扩展性：分布式系统可以通过增加计算机节点来扩展系统的处理能力。

b.可靠性：分布式系统可以通过冗余设计来提高系统的可靠性，即使某个节点发生故障，系统仍然可以正常运行。

c.高性能：分布式系统可以通过并行处理来提高系统的处理速度。

d.灵活性：分布式系统可以根据不同的应用场景进行灵活的配置和部署。

分布式系统的应用领域包括：

a. 大规模数据处理：分布式系统可以用于大规模数据的存储和处理，如 Hadoop、Spark 等。

b. 分布式计算：分布式系统可以用于分布式计算，如分布式机器学习、分布式图计算等。

c.云计算：分布式系统可以用于构建云计算平台，提供云计算服务。

d.分布式存储：分布式系统可以用于构建分布式存储系统，如分布式文件系统、分布式数据库等。

2）常用的 Linux 命令：列举并解释至少 5 条常用的 Linux 命令，包括文件操作、用户管理、网络配置等方面。

ls：用于列出当前目录下的文件和子目录。常用选项包括-l（显示详细信息）、-a（显示所有文件，包括隐藏文件）和-h（以人类可读的方式显示文件大小）。

cp：用于复制文件或目录。常用选项包括-r（递归复制目录及其子目录）、-i（在复制前提示是否覆盖已有文件）和-v（显示复制过程）。

useradd：用于添加新用户。常用选项包括-m（自动创建用户主目录）、-g（指定用户所属的主组）和-s（指定用户默认的 shell）。

passwd：用于修改用户密码。可以直接输入 passwd 命令并按照提示输入新密码，也可以使用选项修改其他用户的密码。

ifconfig：用于配置网络接口。常用选项包括-a（显示所有网络接口）、up（启用指定的网络接口）和 down（禁用指定的网络接口）。ifconfig 也可以用来配置 IP 地址、子网掩码、网关等网络参数。

```
apple@bogon work % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether ac:de:48:00:11:22
        inet6 fe80::aede:48ff:fe00:1122%en7 prefixlen 64 scopeid 0x4
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex>)
        status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 3a:8c:48:f5:d3:c5
        media: autoselect <full-duplex>
        status: inactive
en3: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 3a:8c:48:f5:d3:c1
        media: autoselect <full-duplex>
```

3）正确设置 Linux 的 IP 地址：提供文字说明和截图，解释如何正确设置 Linux 系统的 IP 地址。

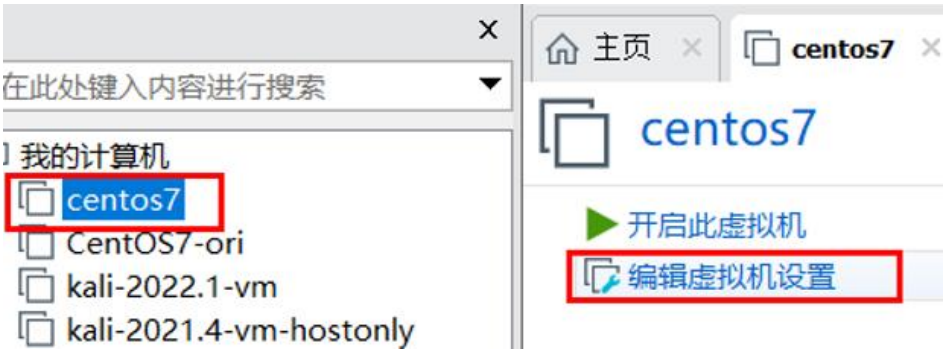
在 Linux 系统中，可以通过命令行或者图形界面来设置 IP 地址。

以下是命令行设置 IP 地址的步骤：

1. 打开终端，输入命令`ifconfig`，查看当前网络接口的信息，如下图所示：

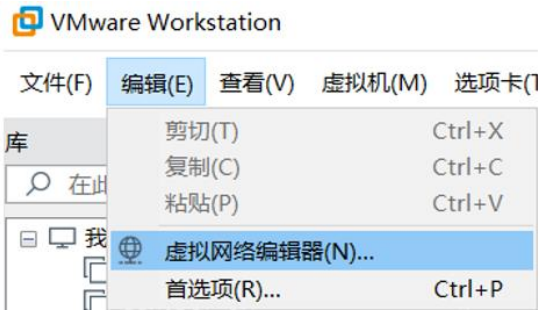
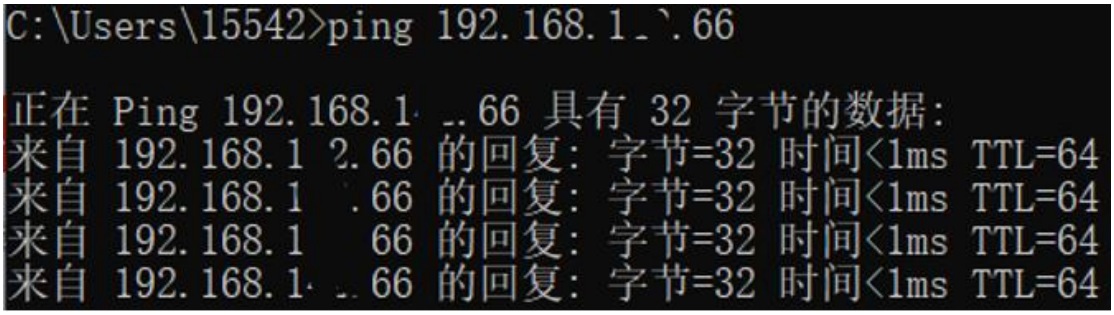
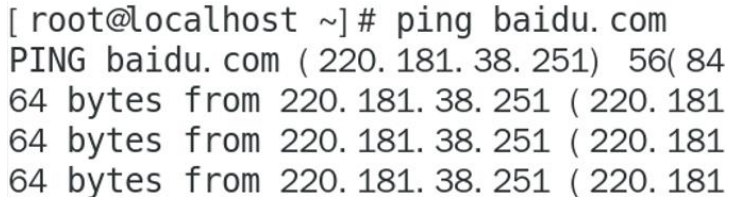
```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Realtek PCIe GbE Family Controller
物理地址. . . . . : 8C-8C-AA-6B-18-E4
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
本地链接 IPv6 地址. . . . . : fe80::b0d6:1a1c:f4de:e329%9(首选)
IPv4 地址 . . . . . : 192.168.3.10(首选)
子网掩码 . . . . . : 255.255.255.0
获得租约的时间 . . . . . : 2022年3月10日 14:08:44
租约过期的时间 . . . . . : 2022年3月12日 17:03:33
默认网关. . . . . : 192.168.3.1
DHCP 服务器 . . . . . : 192.168.3.1
DHCPv6 IAID . . . . . : 109874346
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-27-8C-C9-FF-8C-8C-AA-6B-1
DNS 服务器 . . . . . : 8.8.8.8
TCPIP 上的 NetBIOS . . . . . : 已启用
```

2. 输入命令`sudo nano /etc/network/interfaces`，打开网络配置文件，如下图所示：



3. 在文件中找到需要配置的网络接口，如`eth0`，并添加以下内容：

```
```

auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
```

其中，`address`为需要设置的 IP 地址，`netmask`为子网掩码，`gateway`为网关地址。

4. 保存文件并退出编辑器，输入命令`sudo service networking restart`，重启网络服务。

```
IPADDR="192.168.11.66"
NETMASK="255.255.255.0"
DNS1="114.114.114.114"
GATEWAY="192.168.11.2"
```

5. 输入命令`ifconfig`，查看网络接口的信息，确认 IP 地址已经设置成功，如下图所示：



通过以上步骤，即可以在 Linux 系统中正确设置 IP 地址。



重启测试网络



测试完成

4）环境变量的配置意义：解释环境变量的作用和配置方法，并说明其在 Linux 和 Hadoop 环境中的重要性。

解释：环境变量是一种在操作系统中存储的值，它们可以影响操作系统和应用程序的行为。环境变量的作用是为了方便用户在不同的环境中使用相同的程序或命令，而不需要每次都输入完整的路径或参数。
```

配置方法：

在 Linux 中，环境变量可以通过在 shell 中使用 export 命令来设置。例如，要将 JAVA_HOME 设置为 Java 安装的路径，可以使用以下命令：

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

重要性：在 Hadoop 环境中，环境变量的配置非常重要。Hadoop 是一个分布式系统，由多个节点组成，每个节点都需要访问相同的配置文件和程序。通过设置环境变量，可以确保每个节点都使用相同的配置，从而避免出现不一致的情况。例如，Hadoop 的配置文件中会使用 HADOOP_HOME 环境变量来指定 Hadoop 的安装路径，如果每个节点都设置了相同的 HADOOP_HOME 环境变量，就可以确保每个节点都使用相同的 Hadoop 安装。

总之，环境变量的配置在 Linux 和 Hadoop 环境中都非常重要，它们可以方便地设置和共享配置信息，确保系统和应用程序的一致性和稳定性。

5）Hadoop 的 NameNode 和 DataNode 的作用：解释 NameNode 和 DataNode 在 Hadoop 集群中的职责和功能，并说明其在数据处理中的作用。

解释：

Hadoop 是一个分布式计算框架，它的核心组件包括 HDFS（Hadoop 分布式文件系统）和 MapReduce。在 HDFS 中，NameNode 和 DataNode 是两个重要的组件，它们分别承担着不同的职责和功能。

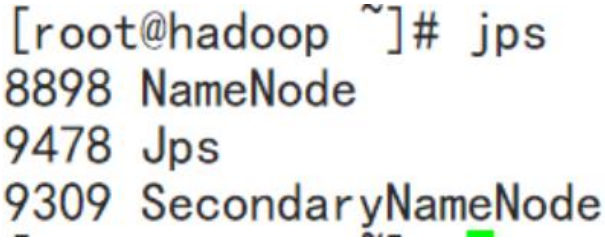
NameNode 是 HDFS 的主节点，它负责管理整个文件系统的命名空间和元数据信息，包括文件名、文件目录结构、文件属性、文件块的位置等。NameNode 还负责协调客户端和 DataNode 之间的数据读写操作，以及监控整个 HDFS 集群的状态和健康状况。因此，NameNode 是 HDFS 的核心组件之一，它的稳定性和可靠性对整个 Hadoop 集群的正常运行至关重要。

DataNode 是 HDFS 的数据节点，它负责存储实际的数据块，并响应客户端和 NameNode 的读写请求。每个 DataNode 都存储着一部分数据块，它们通过心跳机制和周期性的块报告向 NameNode 汇报自己的状态和存储情况。当客户端需要读取数据时，它会向 NameNode 发送请求，NameNode 会返回数据块所在的 DataNode 的地址，客户端再直接从 DataNode 读取数据。当客户端需要写入数据时，它会先向 NameNode 发送请求，NameNode 会返回一组可用的 DataNode 列表，客户端再将数据块分成若干份，分别写入这些 DataNode 中。

思考和总结：

过程错误出现：

启动 HDFS 时用 jps 命令查看，多两个进程：



报以下错误：

```
[root@hadoop hadoop-3.1.3]# start-dfs.sh
Starting namenodes on [hadoop]
ERROR: Attempting to operate on hdfs namenode as root
ERROR: but there is no HDFS_NAMENODE_USER defined. Aborting operation.
Starting datanodes
ERROR: Attempting to operate on hdfs datanode as root
ERROR: but there is no HDFS_DATANODE_USER defined. Aborting operation.
Starting secondary namenodes [hadoop]
ERROR: Attempting to operate on hdfs secondarynamenode as root
ERROR: but there is no HDFS_SECONDARYNAMENODE_USER defined. Aborting operation.
```

解决办法：

在/etc/profile.d/my_env.sh 文件后面添加如下内容：

```
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
```

最终环境变量文件如下：

```
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk1.8.0_212
export PATH=$PATH:$JAVA_HOME/bin
#HADOOP_HOME
export HADOOP_HOME=/opt/module/hadoop-3.1.3
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

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
```

经验教训总结：

以上经历反映了在 Hadoop 集群搭建中环境变量配置的重要性。在启动 HDFS 服务时，由于环境变量设置不正确，导致出现报错。解决方式是在/etc/profile.d/my_env.sh 文件后面添加一些内容，重新配置环境变量来解决问题。这一经历教育我们在 Hadoop 集群搭建中环境变量的设置不容忽视。

在搭建 Hadoop 集群时，环境变量的设置直接影响着集群的正常运行，因此需要认真进行配置。尤其是在多台服务器上搭建 Hadoop 集群时，需要保证所有节点的环境变量设置保持一致，否则很容易出现问题。

为了避免环境变量的配置出现问题，我们可以在每个节点上设置一个统一的/etc/profile.d/目录，然后将环境变量相关的设置脚本存储在该目录下。然后在集群的所有节点上都通过 source 命令来加载该目录下的设置脚本，来保证每个节点的环境变量设置保持一致。

总结反思：

本次实验主要是搭建 Hadoop 分布式系统，通过实验，我对 Hadoop 分布式系统的安装、配置和使用有了更深入的了解。

首先，我们需要安装 Linux 操作系统，我选择了 CentOS 7 作为操作系统。在安装过程中，需要注意一些细节，如分区、网络配置等。安装完成后，我们需要安装 Hadoop 分布式系统和其他必要组件，如 Java、SSH 等。在安装 Hadoop 分布式系统时，需要配置主节点（NameNode）和从节点（DataNode）的配置文件，以及资源管理器（ResourceManager）和节点管理器（NodeManager）的配置文件。在配置过程中，需要注意配置文件的路径和格式，以及各个节点的 IP 地址和端口号等信息。

完成集群配置后，我们需要进行测试和操作。我们使用 wordcount 示例程序对 Hadoop 集群进行测试，该程序可以统计文本中每个单词出现的次数。测试结果显示，Hadoop 集群可以很好地处理大规模数据，并且具有良好的可扩展性和容错性。同时，我们还展示了 Hadoop 集群中关键组件 HDFS、YARN 和 HBase 的 Web UI 界面截图，说明其功能和使用方法。通过这些界面，我们可以方便地管理和监控集群的运行状态，以及查看各个节点的资源使用情况和运行日志等信息。

通过本次实验，我深刻认识到了 Hadoop 分布式系统的重要性和应用价值，也更加熟悉了 Hadoop 分布式系统的安装、配置和使用方法。同时，我也发现了一些问题和不足之处，如配置文件的格式和路径等问题，需要进一步加强学习和实践。