

## 大数据技术-第二章：Hadoop运行及开发环境搭建

### Hadoop完全分布式集群环境搭建（实验2）



# CONTENTS

01. 项目目标

02. 项目描述

03. 知识准备

04. 项目实施

05. 知识拓展

06. 课后实训



# 01

## 项目目标



## 1 项目目标



01

掌握Linux操作系统环境搭建与配置

02

掌握Hadoop完全分布式搭建的步骤

03

掌握Hadoop完全分布式搭建的运行原理

# 02

## 项目描述



## 2 项目描述

Hadoop是一个能够对大量数据进行分布式处理的软件框架，包括并行计算模型Map/Reduce，分布式文件系统HDFS。Hadoop用于解决以下问题：

1. 海量数据的存储(HDFS)
2. 海量数据的分析(MapReduce)
3. 资源管理调度(YARN)



## 2 项目描述

Hadoop部署方式有以下三种：

1. 本地模式：在使用开发工具进行开发调试的时候使用的 只能启动一个map和一个reduce，适用于开发环境。
2. 伪分布式：通过一台机器模拟分布式环境，在开发和学习时使用。适用于测试环境。
3. 集群模式：真实的开发环境。

在大数据行业内Hadoop分布式处理框架主要用于对海量数据存储以及处理。在实际的生产环境内采用该模式，本实验主要是介绍完全分布式环境的搭建，本项目中通过虚拟机新建三台服务器来模拟完全分布式集群的搭建，一台主节点master，两台从节点slave1和slave2。

实验操作系统为CentOS-7-x86\_64-DVD-1511，Hadoop版本为hadoop-2.6.5，JDK版本为jdk-8u201-linux-x64，远程连接工具为SecureCRTPortable，虚拟机工具为VMware-workstation-full-15.5.0-14665864。

本实验所用软件可根据文件名称在资源库内下载。。

# 03

## 知识准备







为了完成本实验，学生需要提前掌握以下理论知识内容：

1. 虚拟机安装部署。
2. Linux运行环境部署。
3. Linux运行环境基础操作。
4. Hadoop安装配置。

# 04

## 项目实施



## 4.1 实施思路

基于项目描述与知识准备的内容，我们已经对Hadoop完全分布式集群环境搭建知识点有了一定的理解，现在我们来基于对完全分布式搭建的理解来学习如何完成完全分布式集群搭建。

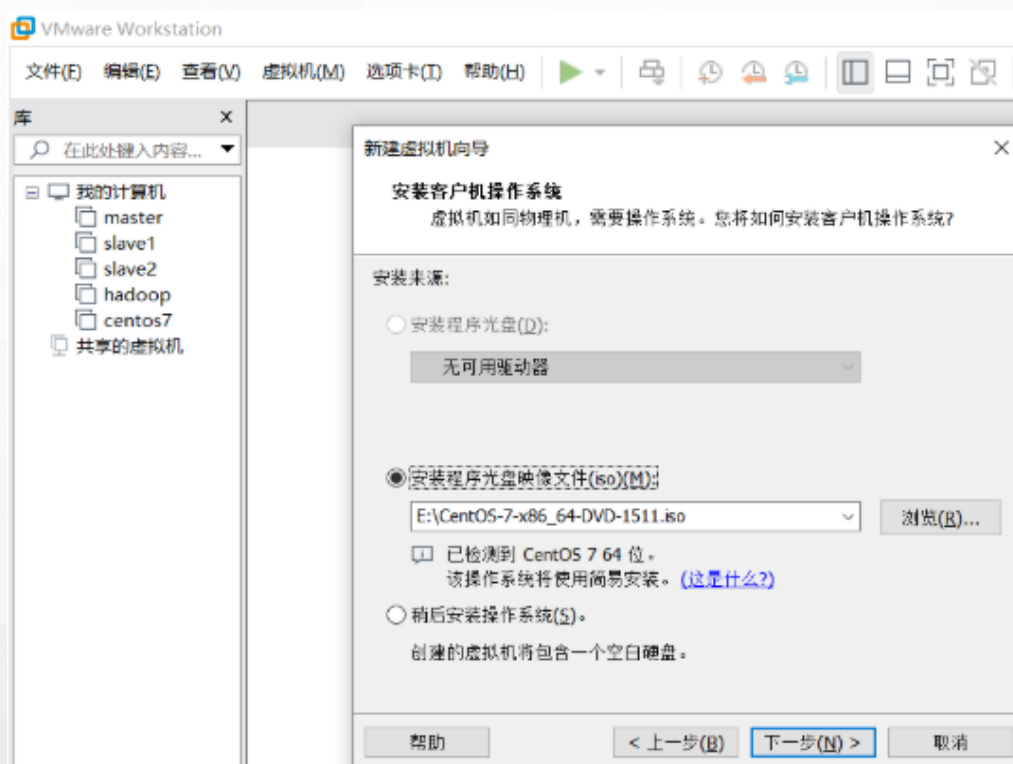
我们将按照下述五个步骤来完成平台的搭建和训练。

1. Linux基础环境部署：在虚拟机中安装Linux操作系统并进行相关的配置，如Linux安装、关闭防火墙、配置YUM源等。
2. JDK安装配置：完成JDK文件上传至系统并完成环境变量配置。
3. Hadoop安装配置：完成Hadoop安装包的上传以及相关配置文件的配置。
4. 克隆master:完成将master克隆到slave1、slave2的操作，并在三台服务器上配置相应服务。
5. 启动集群查看启动进程及验证：通过操作Hadoop平台测试平台的安装的正确性。

## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

使用VMware Workstation Pro新建虚拟机安装CentOS-7-x86\_64-DVD-1511系统,虚拟机配置根据自己本机配置建议设置即可

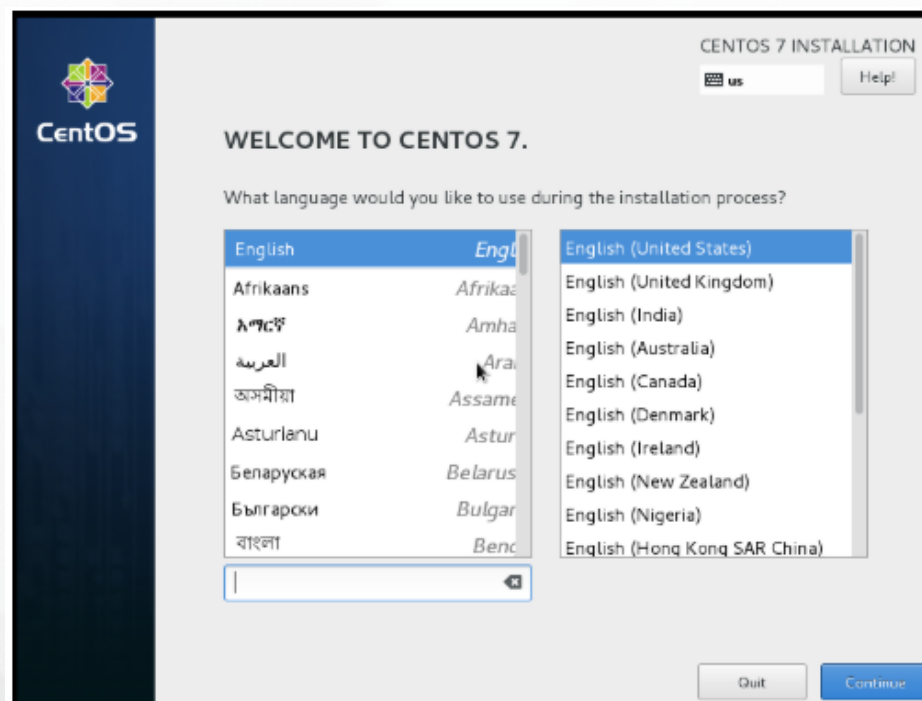




## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

安装过程中选择系统语言为英语



## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

执行如下操作，关闭防火墙

##关闭防火墙

```
[root@localhost ~]# systemctl stop firewalld
```

##永久关闭防火墙

```
[root@localhost ~]# systemctl disable firewalld
```

##查看防火墙状态，验证是否关闭成功

```
[root@localhost ~]# systemctl status firewalld
```

执行结果如下图所示：

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

Feb 23 10:19:41 master systemd[1]: Starting firewalld - dynamic firewall da.....
Feb 23 10:19:42 master systemd[1]: Started firewalld - dynamic firewall daemon.
Feb 23 10:25:54 master systemd[1]: Stopping firewalld - dynamic firewall da.....
Feb 23 10:25:56 master systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

配置网卡，进入/etc/sysconfig/network-scripts目录，根据以下内容编辑ifcfg-eno16777736文件

```
TYPE=Ethernet  
BOOTPROTO=static  
NAME=eno16777736  
UUID=201e5b9d-fb4c-43a4-bf8d-084d47e5f588  
DEVICE=eno16777736  
ONBOOT=yes  
NM_CONTROLLED=yes  
IPADDR=192.168.128.140  
NETMASK=255.255.255.0  
GATEWAY=192.168.128.2  
DNS1=192.168.128.2
```



## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

重启网络服务

```
[root@localhost ~]# service network restart
```

验证配置是否生效，出现下图内容表示配置成功

```
[root@localhost ~]# ip a
```

```
[root@localhost network-scripts]# service network restart
Restarting network (via systemctl): [ OK ]
[root@localhost network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 88:8c:29:46:da:46 brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.148/24 brd 192.168.128.255 scope global eno16777736
        valid_lft forever preferred_lft forever
    inet6 fe08::28c:29ff:fe46:da46/64 scope link
        valid_lft forever preferred_lft forever
[root@localhost network-scripts]#
```

## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

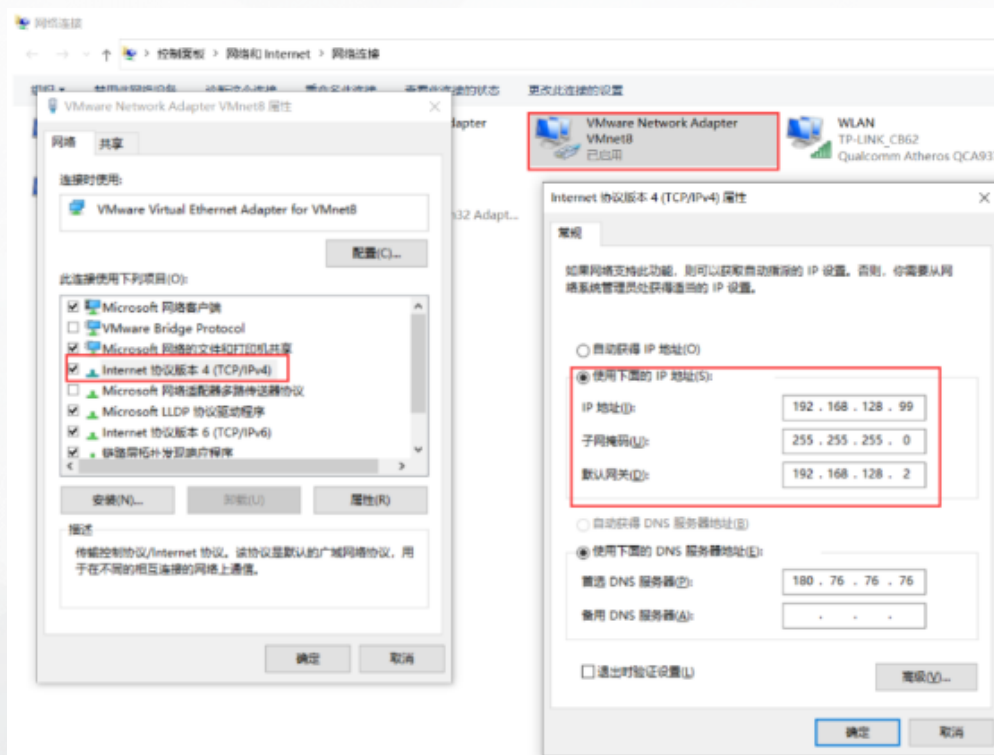
配置远程登录，设置虚拟机适配器信息，在VMware Workstation Pro菜单栏选择编辑，然后再选择虚拟网络编辑器，设置如下图



## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

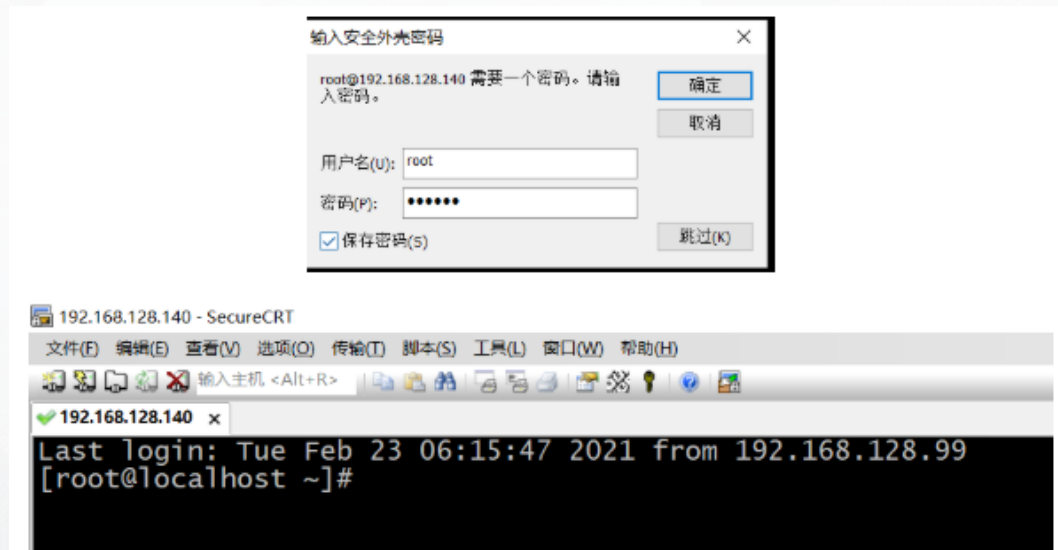
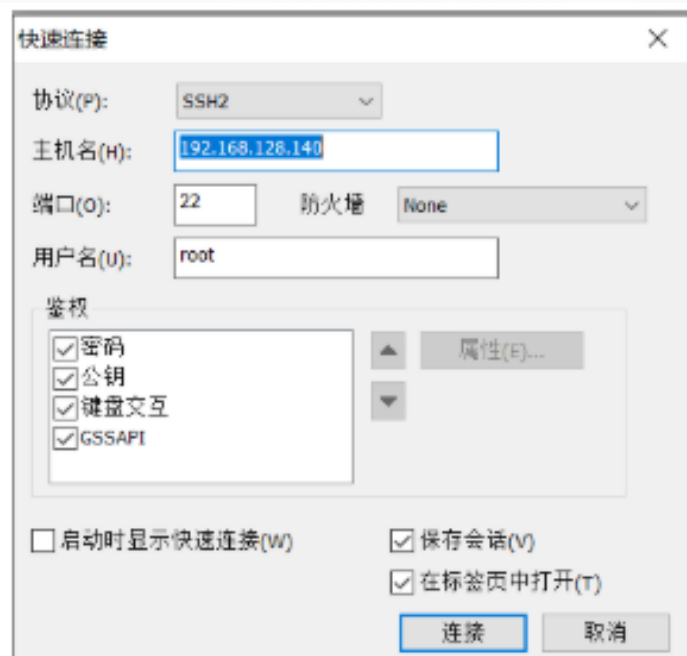
设置本机IP信息，在本机网络和Internet设置中选择VMnet8，然后根据下图设置相应内容，该处要注意IP地址必须与虚拟机网络在同一个网段内



## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

使用SecureCRTPortable远程登录工具连接服务器，在菜单栏选择文件内的快速连接选项填写服务器IP地址点击连接后设置用户名以及密码，点击确定登录服务器，如下图所示





## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

配置YUM源，将CD/DVD勾选已连接，确保设备状态已连接



挂在yum源

```
[root@localhost media]# mount /dev/cdrom /media
mount: /dev/sr0 is write-protected, mounting read-only
```

## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

设置永久挂载

```
#  
# /etc/fstab  
# Created by anaconda on Wed Feb 24 06:46:47 2021  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk'  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info  
#  
/dev/mapper/centos-root / xfs defaults 0 0  
UUID=10299a92-82a1-432c-9a5a-96498f3d8dc4 /boot xfs defaults 0 0  
/dev/mapper/centos-swap swap swap defaults 0 0  
/dev/sr0 /media iso9660 defaults 0 0
```

创建repo文件

```
##yum源备份  
[root@localhost yum.repos.d]# mv /etc/yum.repos.d/* /opt/  
##创建repo文件  
[root@localhost etc]# cd yum.repos.d/  
[root@localhost etc]# vi centos.repo  
[centos]  
name=centos  
baseurl=file:///media  
gpgcheck=0  
enabled=1
```

## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

清除缓存，验证yum源

```
[root@localhost yum.repos.d]# yum clear all
Loaded plugins: fastestmirror
No such command: clear. Please use /usr/bin/yum --help
[root@localhost yum.repos.d]# yum clean all
Loaded plugins: fastestmirror
Cleaning repos: centos
Cleaning up everything
Cleaning up list of fastest mirrors
[root@localhost yum.repos.d]# yum list
```



## 4.2 实施步骤

### 步骤一：安装配置Linux环境：

设置hostname为master

```
[root@localhost ~]# hostnamectl set-hostname master
```

验证是否设置成功

```
[root@master /]# hostnamectl --static
```

Master

重新登录机器名已更改

创建用户

```
[root@master ~]# useradd hadoop
```

```
[root@master ~]# passwd hadoop
```

```
Changing password for user hadoop.
```

```
New password:
```

```
BAD PASSWORD: The password is shorter than 8 characters
```

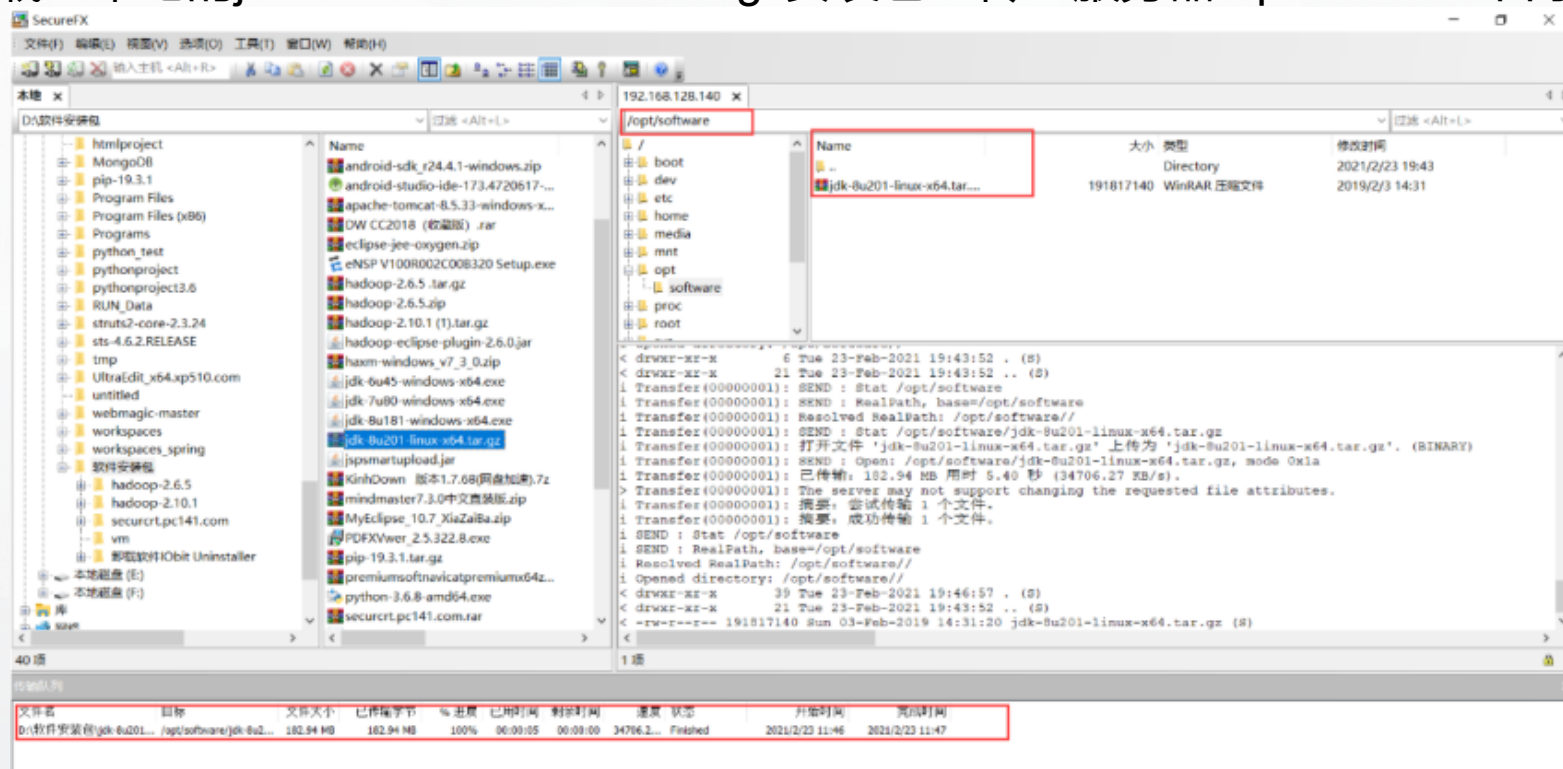
```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

## 4.2 实施步骤

### 步骤二：安装配置JDK：

上传解压JDK文件，使用root用户在服务器/opt目录下新建子目录software并使用SecureFXPortable工具将下载至本地的jdk-8u201-linux-x64.tar.gz安装包上传至服务器/opt/software目录，如下图所示



## 4.2 实施步骤

### 步骤二：安装配置JDK：

将上传后的压缩包进行解压,并更改文件名

```
[root@master software]# tar -zxvf /opt/software/jdk-8u201-linux-x64.tar.gz -C /usr/local/src/  
[root@master src]# mv jdk1.8.0_201/ jdk8
```

配置JDK环境变量，编辑/etc/profile，添加如下内容

```
[root@master ~]# vi /etc/profile  
# JAVA_HOME 指向 JAVA 安装目录  
export JAVA_HOME=/usr/local/src/jdk8  
export PATH=$PATH:$JAVA_HOME/bin # 将 JAVA 安装目录加入 PATH 路径
```

执行 source 使设置生效：

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



## 4.2 实施步骤

### 步骤二：安装配置JDK：

验证安装结果

```
[root@master src]# java -version  
java version "1.8.0_201"  
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)  
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```

能够正常显示 Java 版本则说明 JDK 安装并配置成功。

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

本次安装的是hadoop-2.6.6，使用hadoop用户安装，所以先以hadoop用户登陆。上传并解压Hadoop安装包，安装命令如下，将安装包解压到/usr/local/src/目录下

```
[root@master ~]# tar -zxvf /opt/software/hadoop-2.6.5.tar.gz -C /usr/local/src/
```

修改hadoop-env.sh

```
[hadoop@master ~]$ cd /usr/local/src/hadoop-2.6.5/etc/hadoop/  
[hadoop@master hadoop]$ vi hadoop-env.sh  
##修改export JAVA_HOME=${JAVA_HOME}为:  
export JAVA_HOME=/usr/local/src/jdk8
```

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

修改core-site.xml配置文件

[hadoop@master hadoop]\$ vi core-site.xml

添加如下配置

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/data/hadoopdata</value>
  </property>
</configuration>
```

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

修改hdfs-site.xml配置文件

[hadoop@master hadoop]\$ vi hdfs-site.xml

添加如下配置

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/hadoop/data/hadoopdata/name</value>
    <description>为了保证元数据的安全一般配置多个不同目录</description>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/hadoop/data/hadoopdata/data</value>
    <description>datanode 的数据存储目录</description>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
    <description>HDFS 的数据块的副本存储个数，默认是3</description>
  </property>
</configuration>
```



## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

修改mapred-site.xml配置文件

```
[hadoop@master hadoop]$ cp mapred-site.xml.template mapred-site.xml  
[hadoop@master hadoop]$ vi mapred-site.xml
```

添加如下配置

```
<configuration>  
  <property>  
    <name>mapreduce.framework.name</name>  
    <value>yarn</value>  
  </property>  
</configuration>
```

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

修改yarn-site.xml配置文件

```
[hadoop@master hadoop]$ vi yarn-site.xml
```

添加如下配置

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
  <description>YARN 集群为 MapReduce 程序提供的 shuffle 服务</description>
</property>
</configuration>
```

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

修改slaves配置文件

在slaves文件添加节点机器名master

```
[hadoop@master hadoop]$ vi slaves  
slave1  
slave2
```

配置Hadoop环境变量并使其生效，在文件的最后增加如下两行：

```
[hadoop@master ~]# vi ~/.bashrc
```

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

## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

执行 source 使用设置生效

[hadoop@master ~]# source ~/.bashrc  
验证版本检查设置是否生效，如下图

```
[hadoop@master hadoop]$ hadoop version
Hadoop 2.6.5
Subversion https://github.com/apache/hadoop.git -r e8c9fe0b4c252caf2ebf1464220599650f119997
Compiled by sjlee on 2016-10-02T23:43Z
Compiled with protoc 2.5.0
From source with checksum f05c9fa095a395faa9db9f7ba5d754
This command was run using /usr/local/src/hadoop-2.6.5/share/hadoop/common/hadoop-common-2.6.5.jar
```

出现上述 Hadoop 版本信息就说明 Hadoop 已经安装成功。



## 4.2 实施步骤

### 步骤三：安装配置Hadoop:

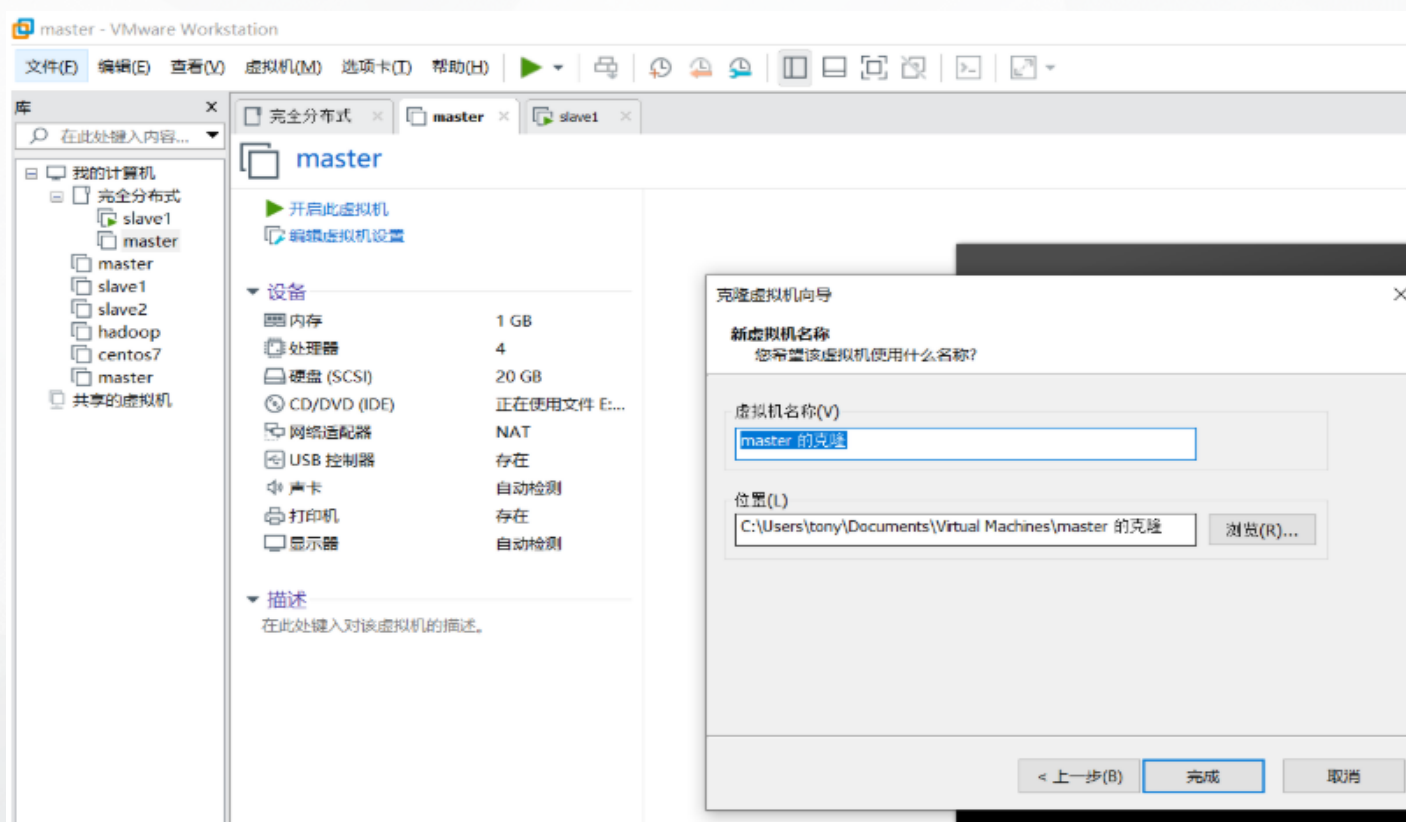
创建hdfs-site.xml里配置的路径

```
[hadoop@master hadoop]$ mkdir -p /home/hadoop/data/hadoopdata/name  
[hadoop@master hadoop]$ mkdir -p /home/hadoop/data/hadoopdata/data
```

## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

选择master点击右键选择管理，然后选择克隆，将master克隆到slave1和slave2，如下图所示



## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

分别登录克隆后的slave1和slave2，修改fcfg-eno16777736网卡配置IP为，192.168.128.142和192.168.128.143，并重启网络服务

```
[root@slave1 network-scripts]# cat ifcfg-eno16777736
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
NAME=eno16777736
UUID=5f3b6da8-4ef6-4a4b-bae6-f302f6cf2f2d
DEVICE=eno16777736
ONBOOT=yes
NM_CONTROLLED=yes
IPADDR=192.168.128.142
NETMASK=255.255.255.0
GATEWAY=192.168.128.2
DNS1=192.168.128.2
```

```
[root@slave1 network-scripts]# sevice network restart
```



## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

在两台机器上分别修改机器名

```
[root@slave1 network-scripts]# hostnamectl set-hostname slave1  
[root@slave2 network-scripts]# hostnamectl set-hostname slave2
```

配置SSH免密码登录，在master上修改hosts文件，并分发至slave1,slave2

```
[root@master ~]# vi /etc/hosts  
192.168.128.141 master  
192.168.128.142 slave1  
192.168.128.143 slave2  
  
##将hadoop1上的/etc/hosts分发到slave1,slave2两个节点  
[root@master ~]# scp /etc/hosts slave1:/etc/hosts  
[root@master ~]# scp /etc/hosts slave2:/etc/hosts
```

## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

在master上生成密钥

```
[hadoop@master .ssh]$ ssh-keygen -t rsa # 三次回车
```

设置本机SSH免密

```
[hadoop@master .ssh]$ ssh-copy-id -i ~/.ssh/id_rsa.pub master # 输入密码
```

将master生成的公匙拷贝到slave1, slave2

```
[hadoop@master .ssh]$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave1 # 输入密码  
[hadoop@master .ssh]$ ssh-copy-id -i ~/.ssh/id_dsa.pub slave2
```

## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

测试是否设置成功

```
##无需输入密码直接进入slave1、slave2，说明免密成功  
[hadoop@master .ssh]$ ssh slave1  
Last login: Wed Feb 24 11:26:28 2021 from master
```

配置NTP服务，统一三台服务器时间,用root用户登录系统，在三台服务器上分别安装NTP服务

```
[root@master ~]# yum install -y ntp  
[root@slave1 ~]# yum install -y ntp  
[root@slave2 ~]# yum install -y ntp
```

## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

在master服务器上编辑/etc/ntp.conf文件,添加以下内容（删除默认server规则）

```
server 127.127.1.0  
fudge 127.127.1.0 stratum 10
```

在master服务器上启动ntp服务

```
[root@master ~]# service ntpd start & chkconfig ntpd on
```



## 4.2 实施步骤

### 步骤四：克隆master到slave1, slave2:

在slave1和slave2分别执行同步命令以及启动服务命令

```
[root@slave1 network-scripts]# ntpdate master
24 Feb 12:07:04 ntpdate[11344]: adjust time server 192.168.128.141 offset 0.410750 sec
[root@slave1 network-scripts]# service ntpd start & chkconfig ntpd on

[root@slave2 ~]# ntpdate master
24 Feb 12:07:25 ntpdate[11292]: adjust time server 192.168.128.141 offset 0.375717 sec
[root@slave2 network-scripts]# service ntpd start & chkconfig ntpd on
```

## 4.2 实施步骤

### 步骤五：启动集群查看启动进程及验证：

在主节点master控制台输入bin/hadoop namenode -format命令格式化namenode

```
[hadoop@master hadoop]$ hadoop namenode -format
```

如图所示出现status 0即为初始化成功。

```
21/02/23 09:35:33 INFO common.Storage: Storage directory /home/hadoop/data/hadoopdata/name has been successfully formatted.
21/02/23 09:35:33 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/data/hadoopdata/name/current/fsimage.ckpt_00000000000000000000 using no compression
21/02/23 09:35:33 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/data/hadoopdata/name/current/fsimage.ckpt_00000000000000000000 of size 323 bytes saved in 0 seconds.
21/02/23 09:35:33 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
21/02/23 09:35:33 INFO util.ExitUtil: Exiting with status 0
21/02/23 09:35:33 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.128.140
*****/
```

## 4.2 实施步骤

### 步骤五：启动集群查看启动进程及验证：

在主机点master控制台输入启动命令：sbin/start-all.sh，启动Hadoop分布式集群

```
[hadoop@master ~]$ cd /usr/local/src/hadoop-2.6.5/  
[hadoop@master hadoop-2.5.6]$ sbin/start-all.sh
```

分别在三台服务器查看启动进程是否成果，运行jps命令，出现以下进程表示启动成功，其中主节点master运行NameNode、ResourceManager、SecondaryNameNode进程服务，两台从节点slave1和slave2运行DataNode、NodeManager进程服务

```
[hadoop@master sbin]$ jps  
11424 SecondaryNameNode  
11251 NameNode  
11561 ResourceManager  
11852 Jps
```

```
[hadoop@slave1 network-scripts]$ jps  
11643 Jps  
11406 DataNode  
11502 NodeManager
```

```
[hadoop@slave2 ~]$ jps  
11602 Jps  
11353 DataNode  
11449 NodeManager
```



## 4.2 实施步骤

### 步骤五：启动集群查看启动进程及验证：

访问WebUI，浏览器访问http://ip:50070，出现如下图页面表示启动成功，查看数据节点数为2表示两台从节点slave1、slave2

Overview 'master:9000' (active)

Started:	Wed Feb 24 12:14:34 EST 2021
Version:	2.6.5, re8c94db4c252caDef1464220599650f1199937
Compiled:	2016-10-02T21:43Z by sjlee from branch-2.6.5
Cluster ID:	CID-859b4b5-adt-4635-85e2-b4e2d0a32ca6
Block Pool ID:	BP-929828731-192.168.128.141-1634186822011

Summary

Security is off.  
Safemode is off.  
1 files and directories, 0 blocks = 1 total filesystem object(s).  
Heap Memory used 14.25 MB of 45.36 MB Heap Memory. Max Heap Memory is 966.69 MB.  
Non Heap Memory used 38.59 MB of 39.38 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	34.92 GB
----------------------	----------

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
slave2 (192.168.128.143:50070)	2	In Service	17.46 GB	4 KB	1.9 GB	15.56 GB	0	4 KB (0%)	0	2.6.5
slave1 (192.168.128.142:50070)	0	In Service	17.46 GB	4 KB	1.9 GB	15.56 GB	0	4 KB (0%)	0	2.6.5

Decommissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks: In files under construction
------	--------------	-------------------------	------------------------------	---



# 05

## 项目拓展

知识点拓展



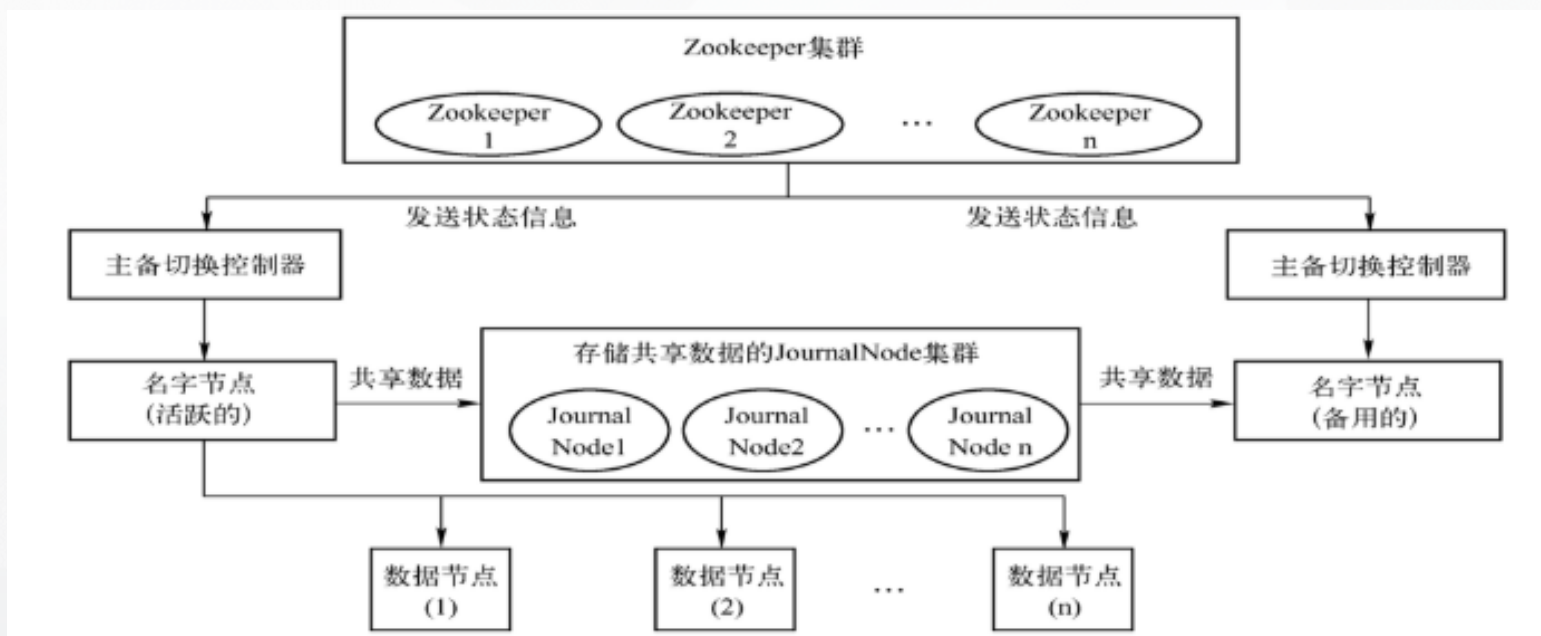
## 5 知识拓展

在前面的任务中，我们学习了如何搭建Hadoop完全分布式集群环境，完全分布式常用于生产环境，而由于完全分布式集群由一个NameNode节点和多个DataNode节点组成，一旦NameNode节点宕机，那么HDFS将不能进行文件的上传与下载，在实际的生产环境下往往会通过高可用技术来规避此类单点故障问题。接下来让我们来了解下HDFS的高可用（拓展知识点）。

HDFS的高可用通常依赖Zookeeper组件，集群内有两个或者两个以上NameNode节点，一个处于Active状态，另一个处于Standby状态。Active NameNode对外提供服务，比如处理来自客户端的RPC请求。而Standby NameNode则不对外提供服务，仅同步Active NameNode的状态，以便能够在它失败时快速进行切换。

## 5 知识拓展

下图展示了HDFS的高可用技术架构



## 5 知识拓展

### 1.NameNode如何进行状态转移

Hadoop提供了FailoverControllerActive和FailoverControllerStandBy两个进程用于监控NameNode的生命周期。

FailoverControllerActive和FailoverControllerStandBy会分别监控对应状态的NameNode，若NameNode无异常则定期向Zookeeper集群发送心跳，若在一定时间内Zookeeper集群没收到FailoverControllerActive发送的心跳，则认为此时状态为Active的NameNode已经无法对外提供服务，因此将状态为StandBy的NameNode切换为Active状态。

### 2.NameNode之间的数据如何进行同步

Hadoop提供了JournalNode用于存放NameNode中的编辑日志，当状态为Active的NameNode节点执行任何名称空间上的修改时，它都会将修改保存到JournalNode集群中，而状态为StandBy的NameNode节点会实时的从JournalNode集群中进行同步。



# 06

## 项目实训

课后练习



## 6.1 实训习题

- 简答题：简述本地模式、伪分布模式、集群模式三种之间的区别以及应用场景？

Turing AI 万维图灵 | 大数据系列课程

大数据

BIG  
DATA

智 / 能 / 科 / 技      放 / 眼 / 未 / 来

