

# 认证考试复习大纲

---

## 笔试部分

---

### 一、选择题

---

1、大数据的4V特征包含（ ）？（选一项）

- A. 大量、单一、高速、价值
- B. 大量、单一、高速、优质
- C. 大量、多样、高速、优质
- D. 大量、多样、高速、价值**

2、下列描述中，错误的是（ ）？（选一项）

- A. Hadoop是一个用于处理大数据的分布式集群架构，支持在GNU/Linux系统以及Windows系统上进行安装使用
- B. SSH是一个软件，专为远程登录会话和其他网络服务提供安全功能的软件**
- C. VMware Workstation是一款虚拟计算机的软件，用户可以在单一的桌面上同时操作不同的操作系统
- D. SecureCRT是一款支持SSH的终端仿真程序，它能够在Windows操作系统上远程连接Linux服务器执行操作

3、部署Hadoop集群时，需要对一些配置文件进行修改，下面列举了5个配置文件：

- ① profile
- ② hadoop-env.sh
- ③ core-site.xml
- ④ ifcfg-eth0
- ⑤ hdfs-site.xml

上面哪些是Hadoop配置文件需要进行修改（ ）？（选一项）

- A. 1、4
- B. 1、3、4
- C. 1、3、5
- D. 2、3、5**

#### 4、MapReduce适用于（ ）？（选一项）

- A、任意可以在Linux上的应用程序
- B、任意应用程序
- C、可以并行处理的应用程序**
- D、可以串行处理的应用程序

#### 5、属于 HDFS架构组成部分是（ ）？（不定项）

- A、NameNode;**
- B、Secondary NameNode;**
- C、DataNode;**
- D、TaskTracker;

#### 6、下列关于MapReduce说法不正确的是（ ）？（选一项）

- A. MapReduce是一种计算框架
- B. MapReduce来源于google的学术论文
- C. MapReduce程序只能用java语言编写**
- D. MapReduce隐藏了并行计算的细节，方便使用

#### 7、Hadoop2.x版本中，默认配置数据块(Block) 的容量大小是（ ）？（选一项）

- A、512M
- B、256M
- C、128M**
- D、64M

8、大数据Hadoop集群中，有关客户端上传文件描述正确的是（ ）。（不定项）

A、数据经过 NameNode 传递给 DataNode

**B、客户端将文件切分为多个Block，依次上传**

**C、客户端发起文件上传请求，通过RPC与NameNode建立通讯**

D、客户端只上传数据到一台 DataNode，然后由 NameNode 负责 Block 复制工作

9、（ ）的输入是排序过的Mapper的输出。（选一项）

**A、Reducer**

B、Mapper

C、Shuffle

D、All of the mentioned

10、在 CentOS 系统中，查看和修改 IP 信息需要用到（ ）文件？（选一项）

A、/etc/resolv.conf

B、/etc/sysconfig/ifcfg-ens33

C、/etc/sysconfig/network/ifcfg-ens33

**D、/etc/sysconfig/network-script/ifcfg-ens33**

11、下面关于MapReduce模型中Map与Reduce函数描述正确的是（ ）。（选一项）

A、Reducee与Reduce之间不是相互独立的

B、Map与Map之间不是相互独立的。

C、一个Map操作就是对每个Reduce所产生的一部分中间结果进行合并操作。

**D、一个Map函数就是对一部分原始数据进行指定的操作。**

## 12、Hadoop配置文件中，不包括（ ）。 （选一项）

- A、core-site.xml
- B、mapred-site.xml
- C、conf-site.xml**
- D、hdfs-site.xml

## 13、关于Secondary NameNode下列哪项描述正确的是（ ）。 （选一项）

- A、它对内存没有要求
- B、它是 NameNode 的热备
- C、SecondaryNameNode 应与NameNode部署到一个节点
- D、它的目的是帮助 NameNode合并编辑日志，减少NameNode启动时间**

## 14、Hadoop的安装部署的模式属于伪分布模式的说法正确的是（ ）。 （选一项）

- A、默认的模式，无需运行任何守护进程（daemon），所有程序都在单个 JVM 上执行
- B、在一台主机模拟多主机，即Hadoop 的守护程序在本地计算机上运行，模拟集群环境，并且是相互独立的Java进程**
- C、完全分布模式的守护进程运行在由多台主机搭建的集群上，是真正的生产环境
- D、高容错全分布模式的守护进程运行在多台主机搭建的集群上

## 15、Zookeeper启动时会最多监听几个端口（ ）？（选一项）

- A、4
- B、3
- C、2**
- D、5

## 16、下列关于Zookeeper描述正确的是（ ）？（不定项）

- A、无论客户端连接的是哪个Zookeeper服务器，其看到的服务端数据模型都是一致的**
- B、从同一个客户端发起的事务请求，最终将会严格按照其发起顺序被应用到zookeeper中
- C、在一个5个节点组成的Zookeeper集群中，如果同时有3台机器宕机，服务不受影响
- D、如果客户端连接到Zookeeper集群中的那台机器突然宕机，客户端会自动切换连接到集群其他机器

## 17、关于 HDFS，说法正确的是（ ）。（选一项）

- A、在HA框架下， NameNode 单点故障不影响HDFS的运行
- B、一个集群可存在多个 NameNode 对外提供服务
- C、不能存储小文件
- D、一个集群可存在多个 DataNode**

## 18、配置Hadoop时，JAVA\_HOME包含在哪一个配置文件中（ ）。（选一项）

- A、hadoop-default.xml
- B、hadoop-env.sh**
- C、hadoop-site.xml
- D、configuration.xsl

## 19、下列选项描述错误的是（ ）。（选一项）

- A、ResourceManager负责的是整个Yarn集群资源的监控、分配和管理工作
- B、Hadoop HA即集群中包含Secondary NameNode作为备份节点存在**
- C、NodeManager负责定时的向ResourceManager汇报所在节点的资源使用情况以及接收并处理来自ApplicationMaster的启动停止容器（Container）的各种请求
- D、初次启动Hadoop HA集群时，需要将格式化文件系统后的目录拷贝至另外一台NameNode节点上

## 20、下面有关Hadoop HA的描述正确的是（ ）。 （不定项）

- A、Hadoop HA是集群中启动两台或两台以上机器充当NameNode，避免一台NameNode节点发生故障导致整个集群不可用的情况
- B、Hadoop HA是两台NameNode同时执行NameNode角色的工作
- C、在Hadoop HA中，Zookeeper集群为每个NameNode都分配了一个故障恢复控制器，该控制器用于监控NameNode的健康状态
- D、ResourceManager是每个节点上的资源和任务管理器

## 21、下列节点数量符合Zookeeper集群部署建议的是（ ）。 （不定项）

- A、2个
- B、3个
- C、4个
- D、5个

## 22、Hive查询语言和SQL的相同之处在于（ ）操作。 （不定项）

- A、Join
- B、Group by
- C、Union
- D、Partition

## 23、以下哪个是Zookeeper集群的角色（ ）？ （不定项）

- A、Leader
- B、Follower
- C、Slave
- D、ObServer

## 24、有关Zookeeper的说法正确的是（ ）？（不定项）

A、Zookeeper对节点的Watch监听通知是永久性的

**B、Zookeeper集群宕机数超过集群数一半，则Zookeeper服务失效**

C、Zookeeper可以作为文件存储系统，因此可以将大规模数据文件存在该系统中

D、部署Zookeeper集群的节点数只能为偶数

## 二、简答题

---

### 1、请列举Hadoop生态体系有哪些（至少五个）？

- HDFS
- MapReduce
- YARN
- Sqoop
- Habase
- Zookeeper
- Hive

### 2、请列举Hadoop集群搭建需要配置的文件（至少五个）？

- hadoop-env.sh
- core-site.xml
- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml
- yarn-env.sh

### 3、请简述Hadoop集群部署方式？

- 独立模式
- 伪分布式模式
- 完全分布式模式

### 4、请描述Zookeeper集群全新集群选举的步骤（3台服务器依次启动）？

步骤一：服务器1启动，先给自己投票然后发投票信息，由于其他机器还没有启动所以它无法接收到投票的反馈信息，因此服务器1的状态一直属于Looking状态。

步骤二：服务器2启动，先给自己投票然后在集群中启动Zookeeper服务的机器发起投票对比，此时与服务器1交换结果，由于服务器2的编号大，所以服务器2胜出，此时服务器1会将票投给服务器2，此时投票数正好大于集群节点半数（ $2 > 3/2$ ），所以服务器2成为领导者状态，服务器1成为追随者状态。

步骤三：服务器3启动，首先，会给自己投票；其次，与之前启动的服务器1、2交换信息，尽管服务器3的编号大，但服务器2已经胜出。所以服务器3只能成为追随者状态。

## 5、请简述Hadoop HA集群初始启动的过程？

步骤一：启动各节点的ZooKeeper组件

步骤二：启动各节点的JournalNode组件

步骤三：格式化第一个NameNode节点，并将该节点生成的格式化临时文件目录，同步更新到其它NameNode节点的数据缓存目录

步骤四：在第一个NameNode节点上，格式化zkfc组件

步骤五：在第一个NameNode节点上，执行start-all.sh。

## 6、假设Hive客户端可成功接入某Hadoop大数据集群操作，创建一个外部表

- emp(员工表)，包含字段 empid、name、sex、address、birthday、deptno、job、salary；
- 给定分隔符为 ','； 给定位置为/hive/emp； (字段中文含义：员工编号、姓名、性别、家庭住址、生日、部门编号、岗位、工资)
- emp表中有数据如下：

101,黎俊杰,男,北京,2000-09-06,A1,测试员,3000

102,刘宏斌,男,香港,1999-10-10,B1,出纳员,4000

103,张彩虹,女,郑州,1990-10-15,A1,开发员,8000

104,郭享,男,天津,1995-08-08,B1,财务,5000

105,蓝海,女,石家庄,1997-05-20,C1,业务部主管,12000

## 按下面要求写出对应HQL查询语句：

### (1) 查询员工总人数

```
select count(empid) empnum from emp;
```



## (2) 查询所有员工工资总额

```
select sum(salary) sum_sal from emp;
```

## (3) 查询工资在3000到5000范围内的员工信息

```
select * from emp where salary between 3000 and 5000;
```

## (4) 查询工资是3000和5000的员工信息

```
select * from emp where salary in (3000,5000);
```

## (5) 查询姓“郭”的员工信息

```
select * from emp where name like '郭%';
```

## (6) 查询姓名中包含“斌”字的员工信息

```
# 方式一
select * from emp where name like '[斌]';
# 方式二
select * from emp where name like '%斌%';
```

# 机试部分

## 一、搭建集群

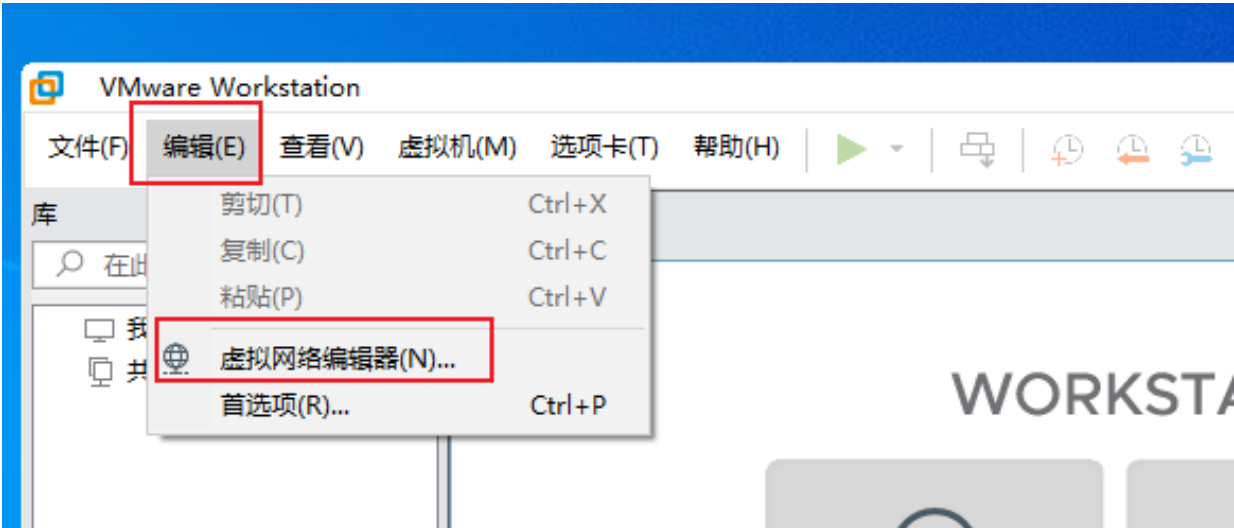
### 1、集群规划说明

主机名	IP信息	用户名/密码	说明
master	192.168.100.128	hadoop	主节点
slave1	192.168.100.129	hadoop	从节点
slave2	192.168.100.130	hadoop	从节点

## 2、集群搭建前期准备

### 2.1、修改虚拟网络信息

#### 2.1.1、打开VMware Workstation工具，编辑--> 虚拟网络编辑器

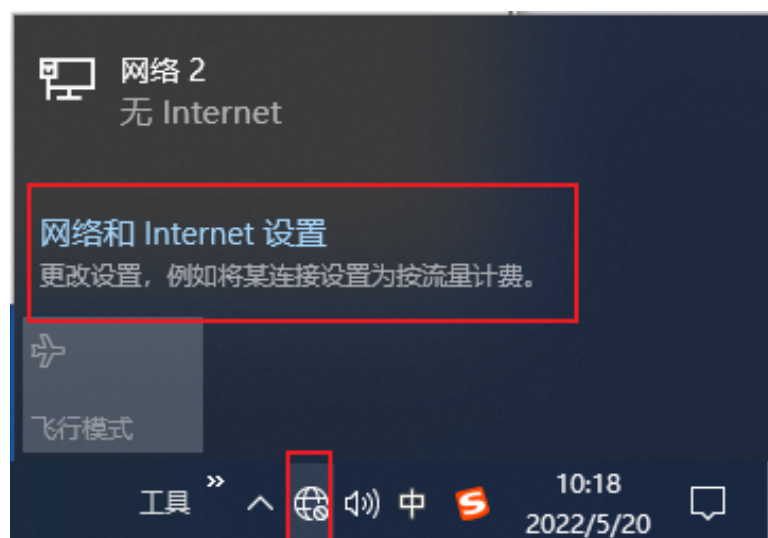


#### 2.1.2、修改VMnet8的子网IP为 192.168.100.0



## 2.2、启用虚拟网络适配器

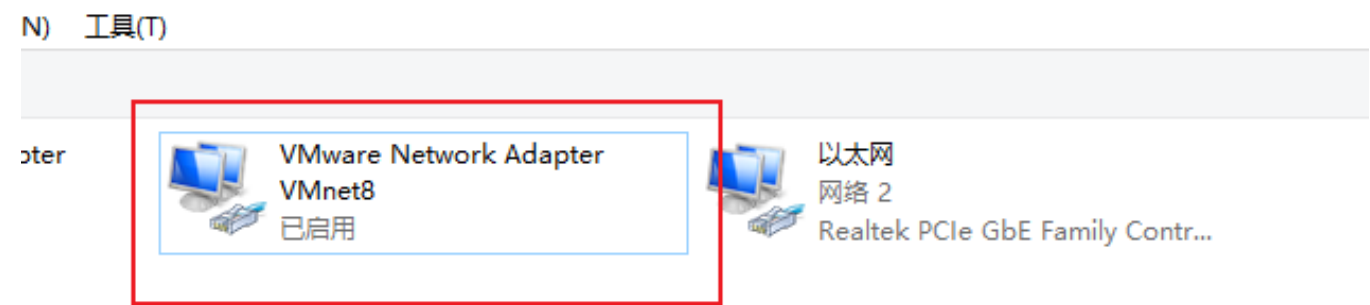
### 2.2.1、点击系统右下角网络图标 --> 网络和Internet设置



2.2.2、点击更改适配器选项

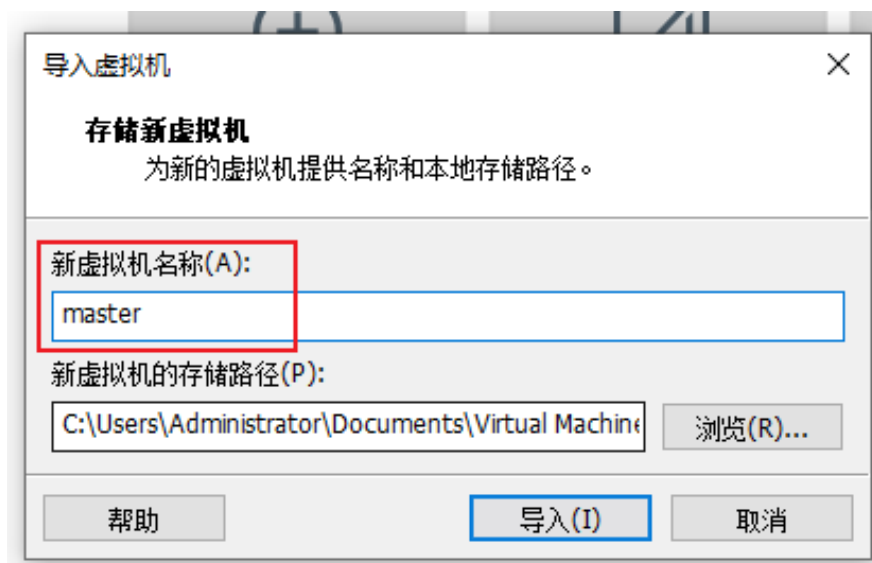
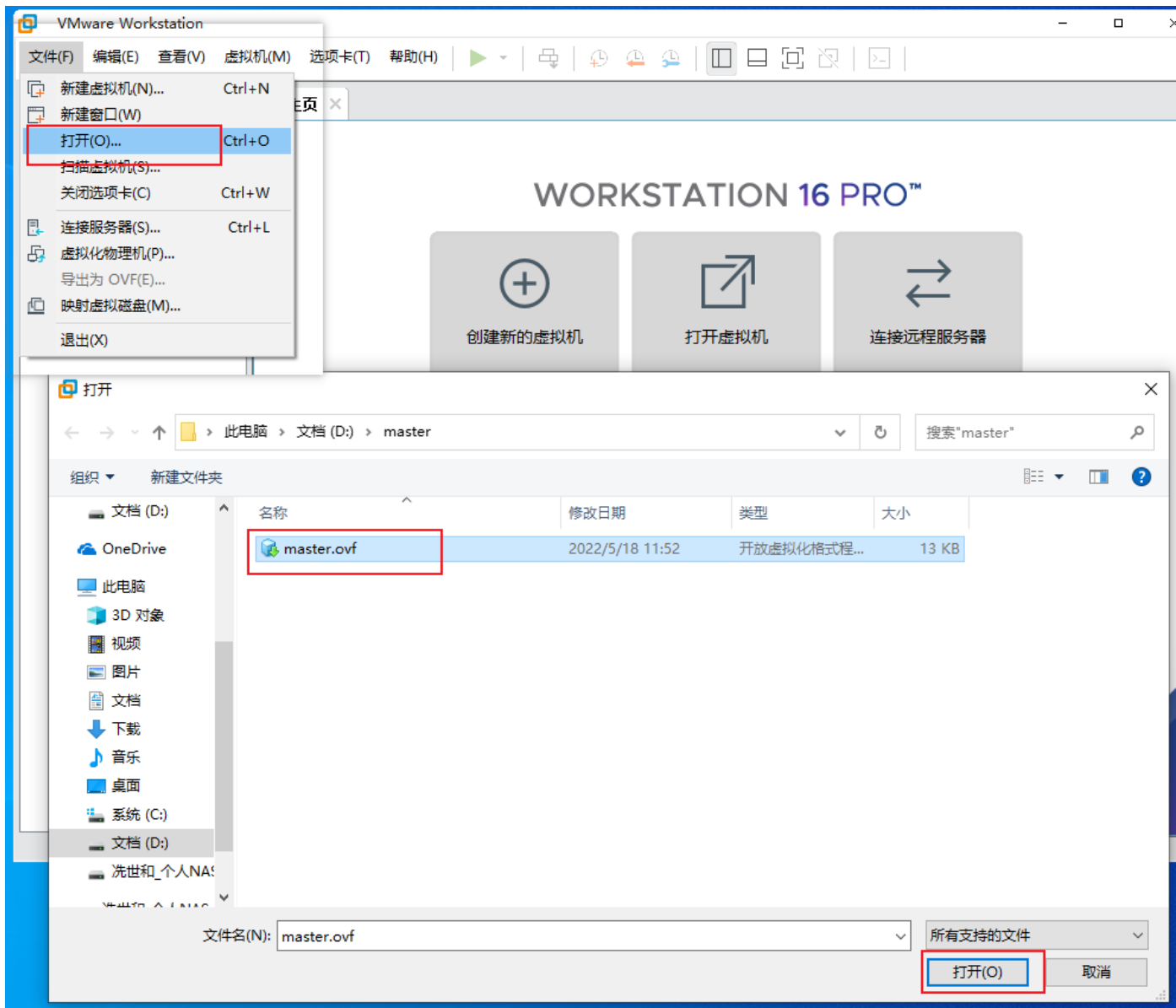


2.2.3、启用VMnet8适配器



3、搭建虚拟机集群

3.1、使用VMware Workstation打开下发的master虚拟机镜像 --> 选择ovf虚拟机镜像文件 --> 导入虚拟机



## 3.2、克隆slave1和slave2节点

### 3.2.1、选择master节点--》鼠标右键--》管理--》克隆--》设置克隆虚拟机名称

克隆虚拟机向导 ×

**新虚拟机名称**  
您希望该虚拟机使用什么名称？

虚拟机名称(V)  
slave1

位置(L)  
C:\Users\Administrator\Documents\Virtual Machines\slave1 浏览(R)...

< 上一步(B) 完成 取消

克隆虚拟机向导 ×

**新虚拟机名称**  
您希望该虚拟机使用什么名称？

虚拟机名称(V)  
slave2

位置(L)  
C:\Users\Administrator\Documents\Virtual Machines\slave2 浏览(R)...

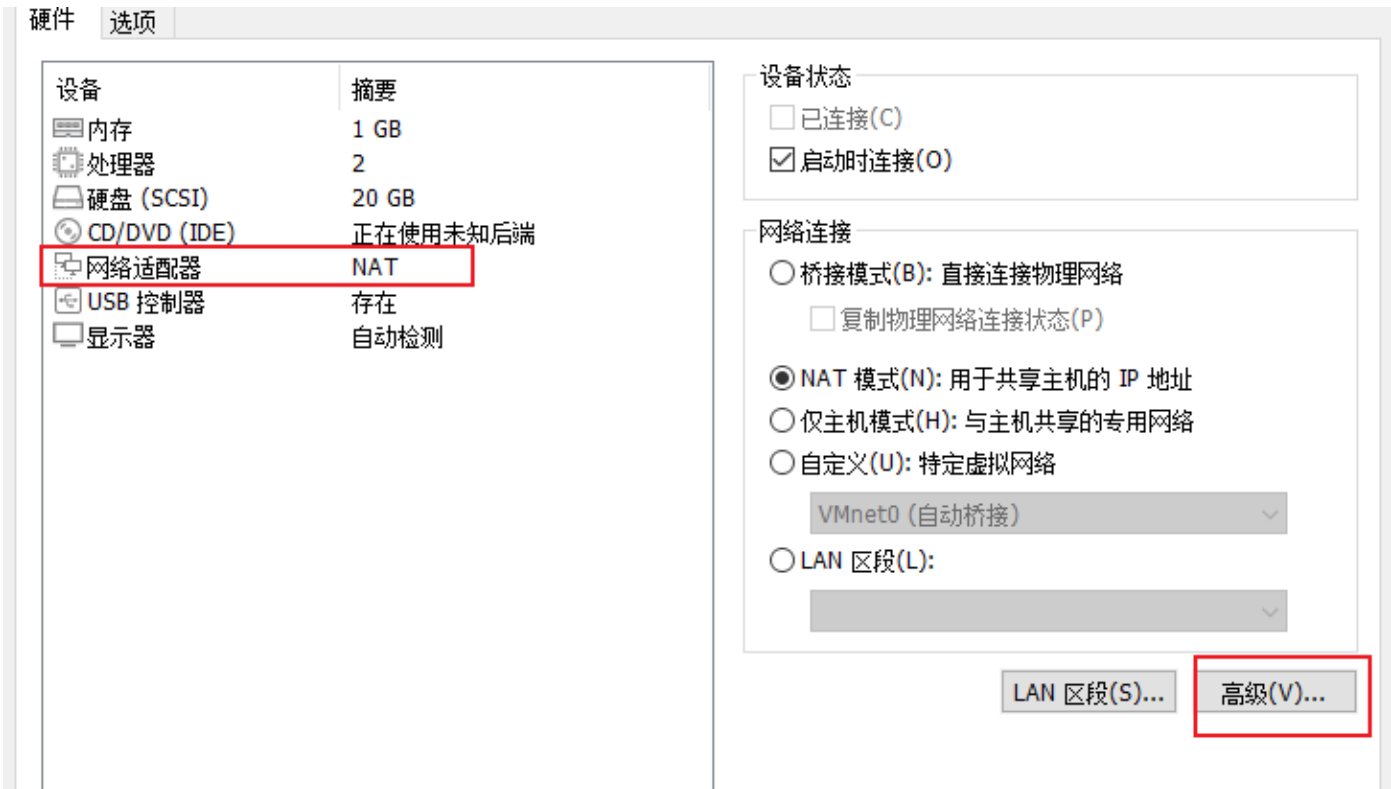
< 上一步(B) 完成 取消

### 3.3、修改三个节点的MAC网卡地址信息

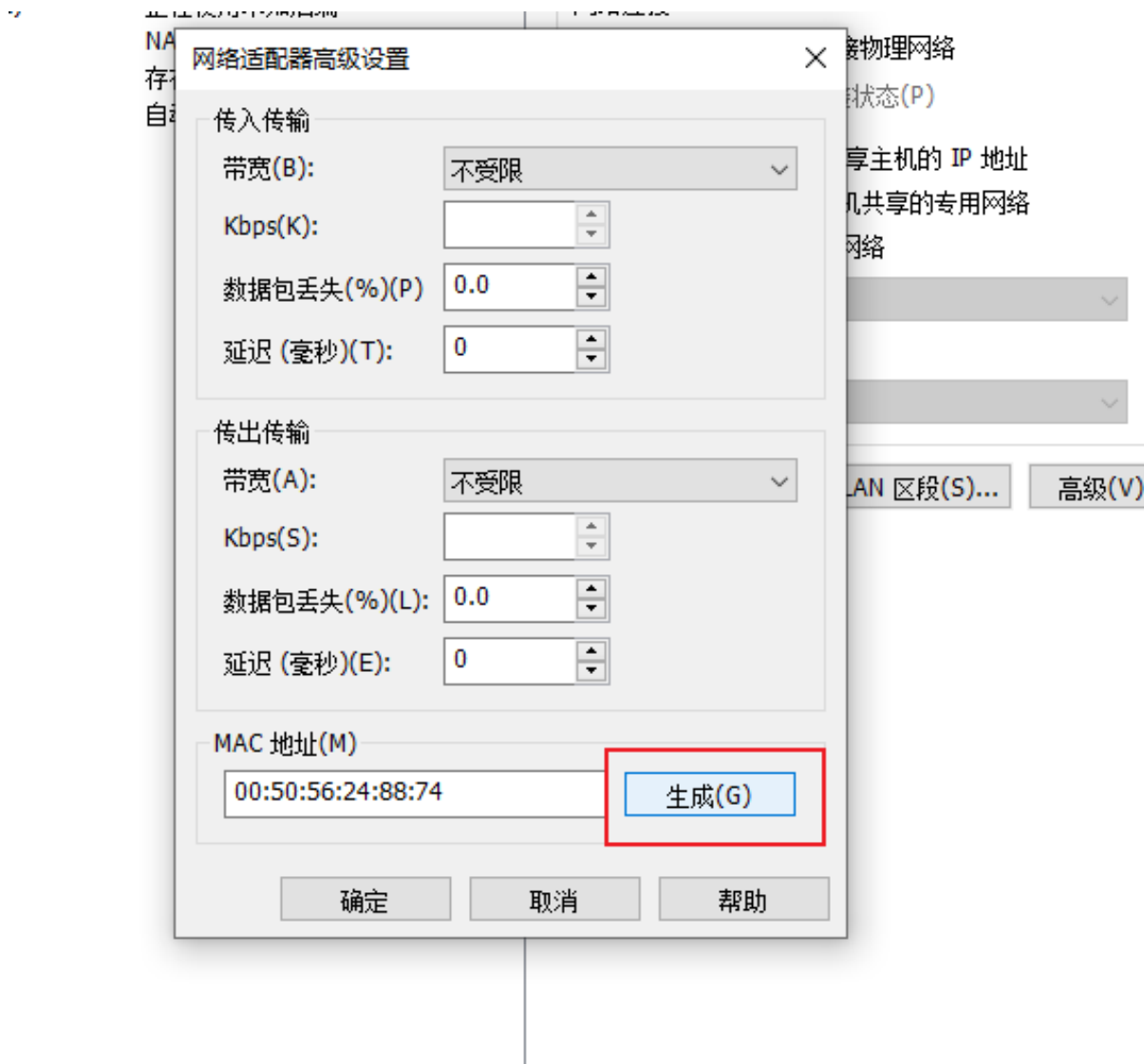
#### 3.3.1、分别选择三台虚拟机--》点击网络适配器



#### 3.3.2、点击高级选项



3.3.3、点击生成选项（可以多点几次）



注意：三台虚拟机都需要生成MAC地址，确保每台虚拟机的网卡信息是唯一的

4、修改虚拟机IP信息

- 说明：
- master节点IP地址为：192.168.100.128
- slave1节点IP地址为：192.168.100.129
- slave2节点IP地址为：192.168.100.130
- 网络信息配置文件路径：`** /etc/sysconfig/network-scripts/ifcfg-ens33 **`



## 4.1、修改master节点IP信息为192.168.100.128

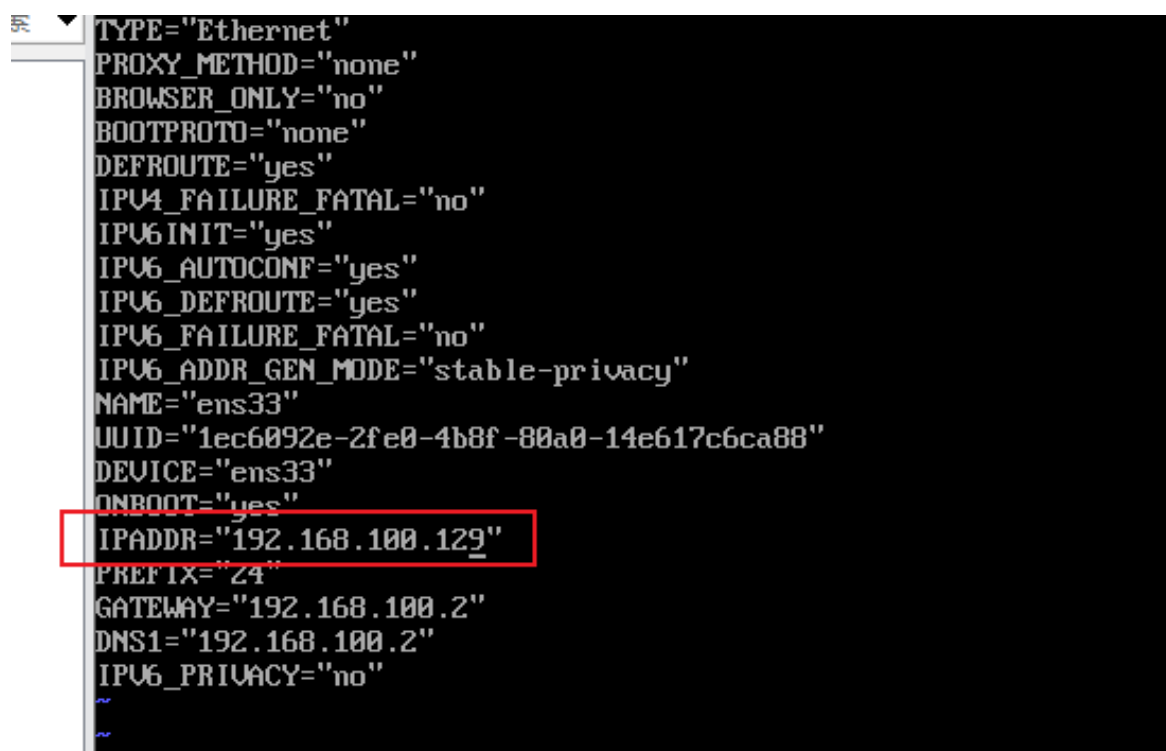
说明：该节点IP信息默认已设为：192.168.100.128，不用修改

## 4.2、修改slave1节点IP信息192.168.100.129

### 4.2.1、使用 vim 编辑器 打开网络信息配置文件；命令如下：

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

### 4.2.2、在配置文件中找到IPADDR选项，将IP修改为129后，保存退出



```
TYPE="Ethernet"  
PROXY_METHOD="none"  
BROWSER_ONLY="no"  
BOOTPROTO="none"  
DEFROUTE="yes"  
IPV4_FAILURE_FATAL="no"  
IPV6_INIT="yes"  
IPV6_AUTOCONF="yes"  
IPV6_DEFROUTE="yes"  
IPV6_FAILURE_FATAL="no"  
IPV6_ADDR_GEN_MODE="stable-privacy"  
NAME="ens33"  
UUID="1ec6092e-2fe0-4b8f-80a0-14e617c6ca88"  
DEVICE="ens33"  
ONBOOT="yes"  
IPADDR="192.168.100.129"  
PREFIX="24"  
GATEWAY="192.168.100.2"  
DNS1="192.168.100.2"  
IPV6_PRIVACY="no"
```

### 4.2.3、重启网卡，命令如下：

```
systemctl restart network
```

**注意：**设置网络后一定要重启网络服务，否则配置无法生效

注意：设置网络后一定要重启网络服务，否则配置无法生效

注意：设置网络后一定要重启网络服务，否则配置无法生效

#### 4.2.4、使用 ip addr 命令查看IP信息是否生效，如果看到IP地址为129则说明修改成功

```
[hadoop@master ~]: ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
    link/ether 00:50:56:2e:c7:74 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.129/24 brd 192.168.100.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::576e:ae0:985e:a185/64 scope link tentative noprefixroute dadfailed
        valid_lft forever preferred_lft forever
    inet6 fe80::ff58:4d9b:bf92:979b/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[hadoop@master ~]:
```

### 4.3、修改slave2节点IP信息192.168.100.130

#### 4.3.1、说明：修改slave2节点IP信息步骤跟slave1节点一样

## 5、更改计算机名和关闭防火墙（master、slave1、slave2）

### 5.1、更改计算机名

#### 5.1.1、修改master节点主机名，命令如下：

```
hostnamectl set-hostname master
```

#### 5.1.2、修改slave1节点主机名，命令如下：

```
hostnamectl set-hostname slave1
```

### 2.1.3、修改slave2节点主机名，命令如下：

```
hostnamectl set-hostname slave2
```

注意：使用命令修改完计算机名需要重启后才生效，重启命令reboot

注意：使用命令修改完计算机名需要重启后才生效，重启命令reboot

注意：使用命令修改完计算机名需要重启后才生效，重启命令reboot

## 2.2、关闭防火墙命令(三个节点都需要执行以下命令)

```
# 关闭防火墙
systemctl stop firewalld.service
# 关闭防火墙自启动
systemctl disable firewalld.service
```

## 6、修改HOSTS文件，完成主机名和IP的绑定

说明：hosts文件路径，`/etc/hosts`

注意：该文件为系统文件，修改时需要加上 `sudo`

注意：三台虚拟机都需要修改该文件，并配置好相应的信息

### 6.1、使用vim编辑器打开hosts文件，命令如下：

```
sudo vim /etc/hosts
```

### 6.2、在该文件中完成主机名和IP的绑定

```
127.0.0.1    localhost localhost.localdomain loca
localhost4  localhost4.localhost4
::1         localhost localhost.localdomain loca
localhost6  localhost6.localhost6

192.168.100.128 master
192.168.100.129 slave1
192.168.100.130 slave2
~
```

## 7、完成SSH免密，实现master与从节点的免密通信

说明：首先在主节点中生成密钥（公钥、私钥），然后将公钥拷贝给从节点

### 7.1、在master节点生成密钥，命令如下：

```
ssh-keygen
```

### 7.2、将公钥拷贝给从节点，命令如下：

```
ssh-copy-id master
ssh-copy-id slave1
ssh-copy-id slave2
```

## 8、解压JDK安装包到“/export/servers/”路径，并配置环境变量及相关文件

### 8.1、解压JDK安装包命令如下：

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /export/servers/
```

## 8.2、配置环境变量

说明：环境变量配置文件路径，`/etc/profile`

注意：该文件是系统文件，修改时需要加上`sudo`

### 8.2.1、使用vim编辑器打开环境变量配置文件，在该文件中添加环境变量

```
export JAVA_HOME=/export/servers/jdk1.8.0_181 # 复制解压后的JDK所在路径
export PATH=$PATH:$JAVA_HOME/bin
```

```
# 配置java环境变量
export JAVA_HOME=/export/servers/jdk1.8.0_181
export PATH=$PATH:$JAVA_HOME/bin
```

### 8.2.2、注意：添加环境变量后需要重新加载环境变量，命令如下

```
source /etc/profile
```

### 8.2.3、查看java版本信息java -version

```
[hadoop@master ~]:java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
[hadoop@master ~]:
```

## 9、安排部署Hadoop集群

说明：将Hadoop安装包解压到`/export/servers/`目录中，并完成相关配置。

注意：Hadoop核心配置参数在下发的“试卷1--附件：试卷所需其它资料”里面

注意：hadoop-env.xml和slaves配置文件需要自己写内容

## 9.1、将Hadoop安装包解压到/export/servers/并设置环境变量

### 9.1.1、解压hadoop安装包命令如下

```
tar -zxvf hadoop-2.7.5.tar.gz -C /export/servers
```

### 9.1.2、添加Hadoop环境变量，使用vim打开/etc/profile文件，在文件末尾处添加以下命令

```
export HADOOP_HOME=/export/servers/hadoop-2.7.5 # 解压后的hadoop目录
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

### 9.1.3、注意：修改配置文件后需要重新加载环境变量，命令如下

```
source /etc/profile
```

### 9.1.4、验证环境变量是否正确，在终端中输入 `hadoop version`，配置成功效果图如下：

```
[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:hadoop version
Hadoop 2.7.5
Subversion Unknown -r Unknown
Compiled by root on 2019-05-19T15:02Z
Compiled with protoc 2.5.0
From source with checksum 9f118f95f47043332d51891e37f736e9
This command was run using /export/servers/hadoop-2.7.5/share/hadoop/common/hadoop-common-2.7.5.jar
[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:
```

## 9.2、修改hadoop集群配置文件

说明：hadoop核心配置文件所在路径：`/export/servers/hadoop-2.7.5/etc/hadoop`

注意：Hadoop集群需要修改以下配置文件

Hadoop运行时配置文件：`hadoop-env.sh`

Hadoop核心配置文件：`core-site.xml`

HDFS核心配置文件：`hdfs-site.xml`

MapReduce核心配置文件：`mapred-site.xml`

YARN核心配置文件：`yarn-site.xml`

集群信息记录文件：`slaves`

注意：`hadoop-env.sh`和`slaves`文件需要自己编写配置信息

注意：`core-site.xml`、`hdfs-site.xml`、`mapred-site.xml`、`yarn-site.xml`配置文件所用到的信息已在“试卷-附件：试卷所需要其它资料”目录中

## 9.2.1、修改hadoop运行时环境配置文件: hadoop-env.sh

```
[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     ssl-client.xml.example
configuration.xml         hdfs-site.xml            kms-site.xml              ssl-server.xml.example
container-executor.cfg    https-env.sh             log4j.properties         yarn-env.cmd
core-site.xml             https-log4j.properties   mapred-env.cmd            yarn-env.sh
hadoop-env.cmd            https-signature.secret   mapred-env.sh             yarn-site.xml
hadoop-env.sh             https-site.xml           mapred-queues.xml.template
hadoop-metrics2.properties kms-acls.xml              mapred-site.xml.template
hadoop-metrics.properties kms-env.sh                slaves
```

说明：使用vim编辑器打开**hadoop-env.sh**，找到`*export JAVA_HOME=${JAVA_HOME}`配置项（大概在文件的25行），将

步骤一：使用vim编辑器打开**hadoop-env.sh**

步骤二：找到 `export JAVA_HOME=${JAVA_HOME}` 配置项（大概在文件的25行）

步骤三：将`${JAVA_HOME}`改为jdk所在位置的绝对路径，如下图

```
24 # The java implementation to use.
25 # export JAVA_HOME=${JAVA_HOME} 原配置
26 export JAVA_HOME=/export/servers/jdk1.8.0_181 修改后配置
27 # 注意：/export/servers/jdk1.8.0_181 为jdk所在位置
```

## 9.2.2、修改 core.site.xml核心配置文件

```
[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     ssl-client.xml.example
configuration.xml         hdfs-site.xml            kms-site.xml              ssl-server.xml.example
container-executor.cfg    https-env.sh             log4j.properties         yarn-env.cmd
core-site.xml             https-log4j.properties   mapred-env.cmd            yarn-env.sh
hadoop-env.cmd            https-signature.secret   mapred-env.sh             yarn-site.xml
hadoop-env.sh             https-site.xml           mapred-queues.xml.template
hadoop-metrics2.properties kms-acls.xml              mapred-site.xml.template
hadoop-metrics.properties kms-env.sh                slaves
```

说明：打开**core.site.xml**配置文件，在文件的 `tags` 标签里面添加以下代码

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://master:9000</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/export/servers/hadoop-2.7.5/tmp</value>
</property>
```

```

19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://master:9000</value>
23   </property>
24   <property>
25     <name>hadoop.tmp.dir</name>
26     <value>/export/servers/hadoop-2.7.5/tmp</value>
27   </property>
28 </configuration>

```

### 9.2.3、修改hdfs-site.xml核心配置文件

```

[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     ssl-client.xml.example
configuration.xml         hdfs-site.xml            kms-site.xml             ssl-server.xml.example
container-executor.cfg    httpfs-env.sh            log4j.properties        yarn-env.cmd
core-site.xml             httpfs-log4j.properties  mapred-env.cmd           yarn-env.sh
hadoop-env.cmd            httpfs-signature.secret  mapred-env.sh            yarn-site.xml
hadoop-env.sh             httpfs-site.xml          mapred-queues.xml.template
hadoop-metrics2.properties kms-acls.xml             mapred-site.xml.template
hadoop-metrics.properties kms-env.sh               slaves

```

说明：打开**hdfs-site.xml**配置文件，在文件的 标签里面添加以下代码

```

<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
<property>
  <name>dfs.namenode.secondary.http-address</name>
  <value>slave1:50090</value>
</property>

```



```

19 <configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>3</value>
23   </property>
24   <property>
25     <name>dfs.namenode.secondary.http-address</name>
26     <value>slave1:50090</value>
27   </property>
28 </configuration>

```

## 9.2.4、修改mapred.site.xml核心配置文件

```

[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     ssl-client.xml.example
configuration.xml         hdfs-site.xml            kms-site.xml              ssl-server.xml.example
container-executor.cfg    httpfs-env.sh            log4j.properties         yarn-env.cmd
core-site.xml             httpfs-log4j.properties  mapred-env.cmd            yarn-env.sh
hadoop-env.cmd            httpfs-signature.secret  mapred-env.sh             yarn-site.xml
hadoop-env.sh             httpfs-site.xml          mapred-queues.xml.template
hadoop-metrics2.properties kms-acls.xml              mapred-site.xml.template
hadoop-metrics.properties kms-env.sh                slaves

```

注意：Hadoop默认只提供**mapred-site.xml.template**模板文件，所以在修改文件前需要拷贝模板文件并重命名为**mapred-site.xml**，命令如下：

```
cp mapred-site.xml.template mapred-site.xml
```

拷贝后效果图如下：

```

[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]:ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     slaves
configuration.xml         hdfs-site.xml            kms-site.xml              ssl-client.xml.example
container-executor.cfg    httpfs-env.sh            log4j.properties         ssl-server.xml.example
core-site.xml             httpfs-log4j.properties  mapred-env.cmd            yarn-env.cmd
hadoop-env.cmd            httpfs-signature.secret  mapred-env.sh             yarn-env.sh
hadoop-env.sh             httpfs-site.xml          mapred-queues.xml.template yarn-site.xml
hadoop-metrics2.properties kms-acls.xml              mapred-site.xml
hadoop-metrics.properties kms-env.sh                mapred-site.xml.template

```

说明：打开**mapred-site.xml**配置文件，在文件的 标签里面添加以下代码

```

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

```

```

19 <configuration>
20     <property>
21         <name>mapreduce.framework.name</name>
22         <value>yarn</value>
23     </property>
24 </configuration>
25

```

### 9.2.5、修改yarn.site.xml核心配置文件

```

[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop] ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     slaves
configuration.xml        hdfs-site.xml            kms-site.xml              ssl-client.xml.example
container-executor.cfg    httpfs-env.sh            log4j.properties         ssl-server.xml.example
core-site.xml             httpfs-log4j.properties  mapred-env.cmd            yarn-env.cmd
hadoop-env.cmd            httpfs-signature.secret  mapred-env.sh            yarn-env.sh
hadoop-env.sh             httpfs-site.xml          mapred-queues.xml.template yarn-site.xml
hadoop-metrics2.properties kms-acls.xml              mapred-site.xml           mapred-site.xml.template
hadoop-metrics.properties kms-env.sh                mapred-site.xml.template

```

说明：打开**yarn.site.xml**配置文件，在文件的 标签里面添加以下代码

```

<property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
</property>
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>

```

```

15 <configuration>
16
17 <!-- Site specific YARN configuration properties -->
18   <property>
19       <name>yarn.resourcemanager.hostname</name>
20       <value>master</value>
21   </property>
22   <property>
23       <name>yarn.nodemanager.aux-services</name>
24       <value>mapreduce_shuffle</value>
25   </property>
26 </configuration>

```

## 9.2.6、修改slaves文件

```

[hadoop@master /export/servers/hadoop-2.7.5/etc/hadoop]: ls
capacity-scheduler.xml    hadoop-policy.xml        kms-log4j.properties     slaves
configuration.xml        hdfs-site.xml            kms-site.xml             ssl-client.xml.example
container-executor.cfg    httpfs-env.sh            log4j.properties        ssl-server.xml.example
core-site.xml            httpfs-log4j.properties  mapred-env.cmd           yarn-env.cmd
hadoop-env.cmd           httpfs-signature.secret  mapred-env.sh           yarn-env.sh
hadoop-env.sh            httpfs-site.xml          mapred-queues.xml.template yarn-site.xml
hadoop-metrics2.properties kms-acls.xml             mapred-site.xml
hadoop-metrics.properties kms-env.sh              mapred-site.xml.template

```

说明：打开**slaves**配置文件，将文件中原有内容删除并添加以下代码

```

master
slave1
slave2

```

```

1 master
2 slave1
3 slave2

```

## 9.3、同步集群配置文件到从节点

说明：由于集群中各节点都需要安排和配置jdk和hadoop等服务，因此只需要将master节点中已经配置好的文件分发到从节点中即可。

### 9.3.1、同步系统环境变量文件，将环境变量配置文件/etc/profile文件分发到从节点中

注意：该文件属于系统文件，分发时需要加上sudo

```
# 分发到slave1节点
sudo scp /etc/profile slave1:/etc/profile
# 分发到slave2节点
sudo scp /etc/profile slave2:/etc/profile
```

```
[hadoop@master ~]:sudo scp /etc/profile slave1:/etc/profile
[sudo] hadoop 的密码: 命令
The authenticity of host 'slave1 (192.168.100.129)' can't be established.
ECDSA key fingerprint is SHA256:sDxAueCh1S0/V3tbfkvImHiCFYWFN0/Y/n6fy1E3kVA.
ECDSA key fingerprint is MD5:1f:db:c5:48:b1:20:06:ad:29:dc:5f:12:1e:da:63:c1.
Are you sure you want to continue connecting (yes/no)? yes 输入: yes
Warning: Permanently added 'slave1,192.168.100.129' (ECDSA) to the list of known hosts.
root@slave1's password: 输入密码
profile 结果状态 100% 2054 2.9MB/s 00:00
[hadoop@master ~]:
```

### 9.3.2、同步jdk文件，将master节点中的的jdk文件分发到从节点中

说明：将master节点中的/export/servers/jdk1.8.0\_181目录拷贝到slave1和slave2从节点中，命令如下：

```
# 拷贝给slave1节点
scp -r /export/servers/jdk1.8.0_181 slave1:/export/servers
# 拷贝给slave2节点
scp -r /export/servers/jdk1.8.0_181 slave2:/export/servers
```

注意：在分发过程中可能会询问是否继续连接，一定要输入yes，如下图：

```
[hadoop@master ~]:scp -r /export/servers/jdk1.8.0_181 slave1:/export/servers
The authenticity of host 'slave1 (192.168.100.129)' can't be established.
ECDSA key fingerprint is SHA256:sDxAueCh1S0/V3tbfkvImHiCFYWFN0/Y/n6fy1E3kVA.
ECDSA key fingerprint is MD5:1f:db:c5:48:b1:20:06:ad:29:dc:5f:12:1e:da:63:c1.
Are you sure you want to continue connecting (yes/no)? yes 询问是否继续连接，输入yes
Warning: Permanently added 'slave1,192.168.100.129' (ECDSA) to the list of known hosts.
hadoop@slave1's password: 输入hadoop密码
```

### 9.3.3、同步将hadoop安装包到从节点中

说明：将master节点中的/export/servers/hadoop-2.7.5目录拷贝到slave1和slave2从节点中，命令如下：

```
# 分发到slave1节点
scp -r /export/servers/hadoop-2.7.5 slave1:/export/servers
# 分发到slave2节点
scp -r /export/servers/hadoop-2.7.5 slave2:/export/servers
```

## 9.4、刷新所有节点/etc/profile配置文件，重新加载系统环境变量

说明：由于各节点的环境变量配置文件已经做了修改，所以需要重新加载环境变量以便jdk和hadoop环境能够生效；命令如下：

注意，每个节点都需要执行

```
source /etc/profile
```

## 9.5、设置hadoop目录权限

说明：为了防止hadoop在使用过程中出现权限问题，建议在每个节点中设置hadoop目录权限，命令如下：

```
sudo chmod -R 777 /export/
```

## 9.6、格式化集群

在主节点（master）中使用以下命令对hadoop集群进行初始化

```
hadoop namenode -format
```

## 9.7、启动hadoop集群

在主节点（master）中使用 **start-all.sh**命令一键启动集群

```
start-all.sh
```

### 9.7.1、检查各节点hadoop服务运行情况

说明：在每个节点中使用jps查看hadoop运行情况，

- master节点截图

```
[hadoop@master ~]:jps
97651 DataNode
97481 NameNode
98152 NodeManager
98426 Jps
97996 ResourceManager
[hadoop@master ~]:
```

- slave1节点截图

```
[hadoop@slave1 ~]:jps
96149 SecondaryNameNode
96485 Jps
96279 NodeManager
95998 DataNode
[hadoop@slave1 ~]:
```

- slave2节点截图

```
[hadoop@slave2 ~]:jps
95705 DataNode
95929 NodeManager
96200 Jps
[hadoop@slave2 ~]:
```

## 10、安装部署zookeeper服务

说明：将zookeeper安装包解压到/export/servers/目录中，并完成相关配置。

## 10.1、将zookeeper安装包解压到/export/servers/并设置环境变量

### 10.1.1、解压zookeeper安装包命令如下

```
tar -zxvf zookeeper-3.4.10.tar.gz -C /export/servers/
```

### 10.1.2、添加zookeeper环境变量，使用vim打开/etc/profile文件，在文件末尾处添加以下命令

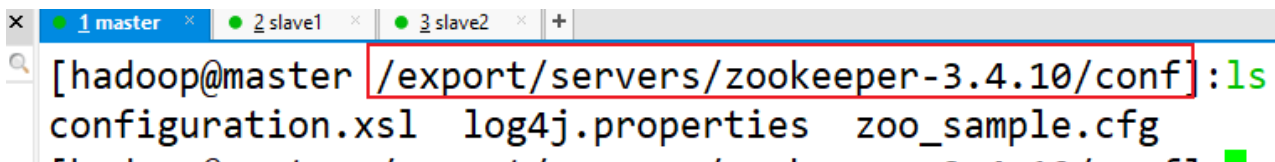
```
export ZK_HOME=/export/servers/zookeeper-3.4.10 # 解压后的zookeeper路径
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$ZK_HOME/bin
```

## 10.2、修改zookeeper服务配置文件

说明：zookeeper核心配置文件所在路径：`/export/servers/zookeeper-3.4.10/conf`

注意：zookeeper服务需要修改以下配置文件

zookeeper核心配置文件：`zoo.cfg`

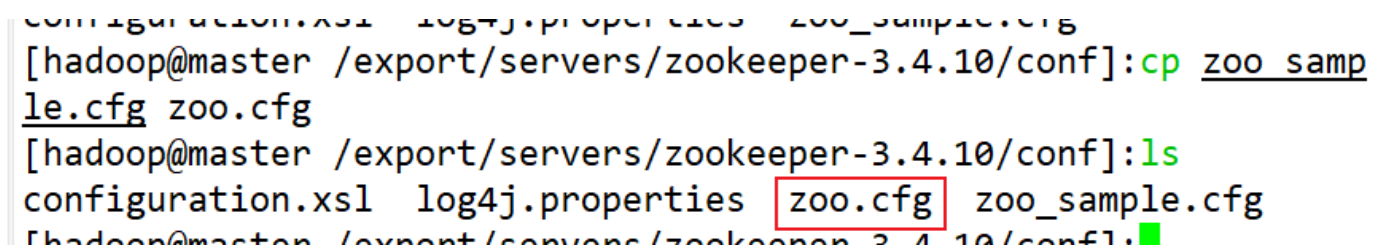


```
[hadoop@master /export/servers/zookeeper-3.4.10/conf]:ls
configuration.xml  log4j.properties  zoo_sample.cfg
```

说明：由于zookeeper默认只提供`zoo_sample.cfg`文件，因此需要将`zoo_sample.cfg`拷贝并重命名为`zoo.cfg`，命令如下：

```
cp zoo_sample.cfg zoo.cfg
```

### 10.2.1、修改zookeeper配置文件: zoo.cfg



```
configuration.xml  log4j.properties  zoo_sample.cfg
[hadoop@master /export/servers/zookeeper-3.4.10/conf]:cp zoo_sample.cfg zoo.cfg
[hadoop@master /export/servers/zookeeper-3.4.10/conf]:ls
configuration.xml  log4j.properties  zoo.cfg  zoo_sample.cfg
```

步骤一：使用vim编辑器打开`zoo.cfg`

步骤二：找到 `dataDir=/tmp/zookeeper` 配置项（大概在文件的12行）

步骤三：将`dataDir=`配置项的值改为：`/export/data/zookeeper/zkdata`，如下图

```
12 # dataDir=/tmp/zookeeper 原路径
13 dataDir=/export/data/zookeeper/zkdata 修改后路径
```

步骤四：在zoo.cfg文件的末尾处添加以下代码

```
# zookeeper集群服务器编码对应的主机名，通信端口号
server.1=master:2888:3888
server.2=slave1:2888:3888
server.3=slave2:2888:3888
```

## 10.3、在/export/data/目录下创建zookeeper/zkdata目录

```
mkdir -p /export/data/zookeeper/zkdata
```

## 10.4、创建myid文件

说明：在/export/data/zookeeper/zkdata目录中创建myid文件，并在myid文件中记录zookeeper服务器编号。

master节点的服务器编号为：1

slave1节点的服务器编号为：2

slave2节点的服务器编号为：3

步骤一：使用vim编辑器新建并打开myid文件，命令如下：

```
vim /export/data/zookeeper/zkdata/myid
```

步骤二：在myid文件中添加master服务器编号：1

## 10.5、同步集群配置文件到从节点

说明：由于集群中各节点都需要安排和配置jdk和hadoop、zookeeper等服务，因此只需要将master节点中已经配置好的文件分发到从节点中即可。

### 10.5.1、同步系统环境变量文件，将环境变量配置文件/etc/profile文件分发到从节点中

注意：该文件属于系统文件，分发时需要加上sudo

```
# 分发到slave1节点
sudo scp /etc/profile slave1:/etc/profile
# 分发到slave2节点
sudo scp /etc/profile slave2:/etc/profile
```



```
[hadoop@master ~]:sudo scp /etc/profile slave1:/etc/profile
[sudo] hadoop 的密码: 命令
The authenticity of host 'slave1 (192.168.100.129)' can't be established.
ECDSA key fingerprint is SHA256:sDxAueCh1S0/V3tbfkvImHiCFYWFN0/Y/n6fy1E3kVA.
ECDSA key fingerprint is MD5:1f:db:c5:48:b1:20:06:ad:29:dc:5f:12:1e:da:63:c1.
Are you sure you want to continue connecting (yes/no)? yes 输入: yes
Warning: Permanently added 'slave1,192.168.100.129' (ECDSA) to the list of known hosts.
root@slave1's password: 输入密码
profile 结果状态 100% 2054 2.9MB/s 00:00
[hadoop@master ~]:
```

### 10.5.2、同步将zookeeper安装包到从节点中

说明：将master节点中的/export/servers/zookeeper-3.4.10目录拷贝到slave1和slave2从节点中，命令如下：

```
# 分发到slave1节点
scp -r /export/servers/zookeeper-3.4.10 slave1:/export/servers
# 分发到slave2节点
scp -r /export/servers/zookeeper-3.4.10 slave2:/export/servers
```

### 10.5.3、同步将zookeeper数据目录到从节点中

说明：将master节点中的/export/data/zookeeper目录拷贝到slave1和slave2从节点中，命令如下：

```
# 分发到slave1节点
scp -r /export/data/zookeeper slave1:/export/data/
# 分发到slave2节点
scp -r /export/data/zookeeper slave2:/export/data/
```

### 10.5.4、修改slave1和slave2节点的myid文件

步骤一：修改slave1节点的myid值为：2

步骤二：修改slave2节点的myid值为：3

## 10.6、刷新所有节点/etc/profile配置文件，重新加载系统环境变量

说明：由于各节点的环境变量配置文件已经做了修改，所以需要重新加载环境变量以便jdk和hadoop及zookeeper环境能够生效；命令如下：

注意，每个节点都需要执行

```
source /etc/profile
```

## 10.7、重新设置/export/目录权限

由于各集群在启动时会对目录进行操作，为了避免出现权限问题，建议给指定目录赋予最高权限

注意：三个节点都需要设置权限

命令如下：

```
# master节点
sudo chmod -R 777 /export/
# slave1节点
sudo chmod -R 777 /export/
# slave2节点
sudo chmod -R 777 /export/
```

## 11、启动zookeeper集群

说明：由于zookeeper集群在启动时会有一个投票选举过程，因此在启动集群时要依次启动

启动顺序：master --> slave1 --> slave2

### 11.1、master节点启动zookeeper服务命令

```
zkServer.sh start
```

```
[sudo] hadoop 的密码:
[hadoop@master /export/data/zookeeper/zkdata]: zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg
g
Starting zookeeper ... STARTED
[hadoop@master /export/data/zookeeper/zkdata]:
```

### 11.2、slave1节点启动zookeeper服务命令

```
zkServer.sh start
```

```
[sudo] hadoop 的密码:
[hadoop@master /export/data/zookeeper/zkdata]: zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg
g
Starting zookeeper ... STARTED
[hadoop@master /export/data/zookeeper/zkdata]:
```

## 11.3、slave2节点启动zookeeper服务命令

```
zkServer.sh start
```

```
[suoo] hadoop 的密码:  
[hadoop@master /export/data/zookeeper/zkdata]: zkServer.sh start  
ZooKeeper JMX enabled by default  
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg  
Starting zookeeper ... STARTED  
[hadoop@master /export/data/zookeeper/zkdata]:
```

## 12、查看zookeeper集群选举结果

zookeeper集群在启动时会进行投票选举，并为每个服务器设置一个角色，集群启动完成后可以通过命令查看服务器的角色信息

### 12.1、查看master节点zookeeper角色

```
zkServer.sh status
```

```
[hadoop@master ~]: zkServer.sh status  
ZooKeeper JMX enabled by default  
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg  
Mode: follower  
[hadoop@master ~]:
```

### 12.2、查看slave1节点zookeeper角色

```
zkServer.sh status
```

```
[hadoop@slave1 ~]: zkServer.sh status  
ZooKeeper JMX enabled by default  
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg  
Mode: leader  
[hadoop@slave1 ~]:
```

## 12.3、查看slave2节点zookeeper角色

```
zkServer.sh status
```

```
[hadoop@slave2 ~]:zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /export/servers/zookeeper-3.4.10/bin/../conf/zoo.cfg
Mode: follower
[hadoop@slave2 ~]:
```

## 13、查看QuorumPeerMain进程信息

说明：zookeeper集群启动服务后会自动开启QuorumPeerMain进程，可通过jps命令查看hadoop和zookeeper服务进程

### 13.1、查看master节点QuorumPeerMain进程信息

```
jps
```

```
[hadoop@master ~]:
[hadoop@master ~]:jps
5347 QuorumPeerMain
4519 DataNode
4392 NameNode
5450 Jps
4763 ResourceManager
4877 NodeManager
[hadoop@master ~]:
```

### 13.1、查看slave1节点QuorumPeerMain进程信息

```
jps
```

```
[hadoop@slave1 ~]:jps
2272 NodeManager
2656 QuorumPeerMain
2739 Jps
2204 SecondaryNameNode
2110 DataNode
[hadoop@slave1 ~]:
```

### 13.1、查看slave2节点QuorumPeerMain进程信息

jps

```
[hadoop@slave2 ~]:jps
2449 NodeManager
2661 QuorumPeerMain
2725 Jps
2349 DataNode
[hadoop@slave2 ~]:
```

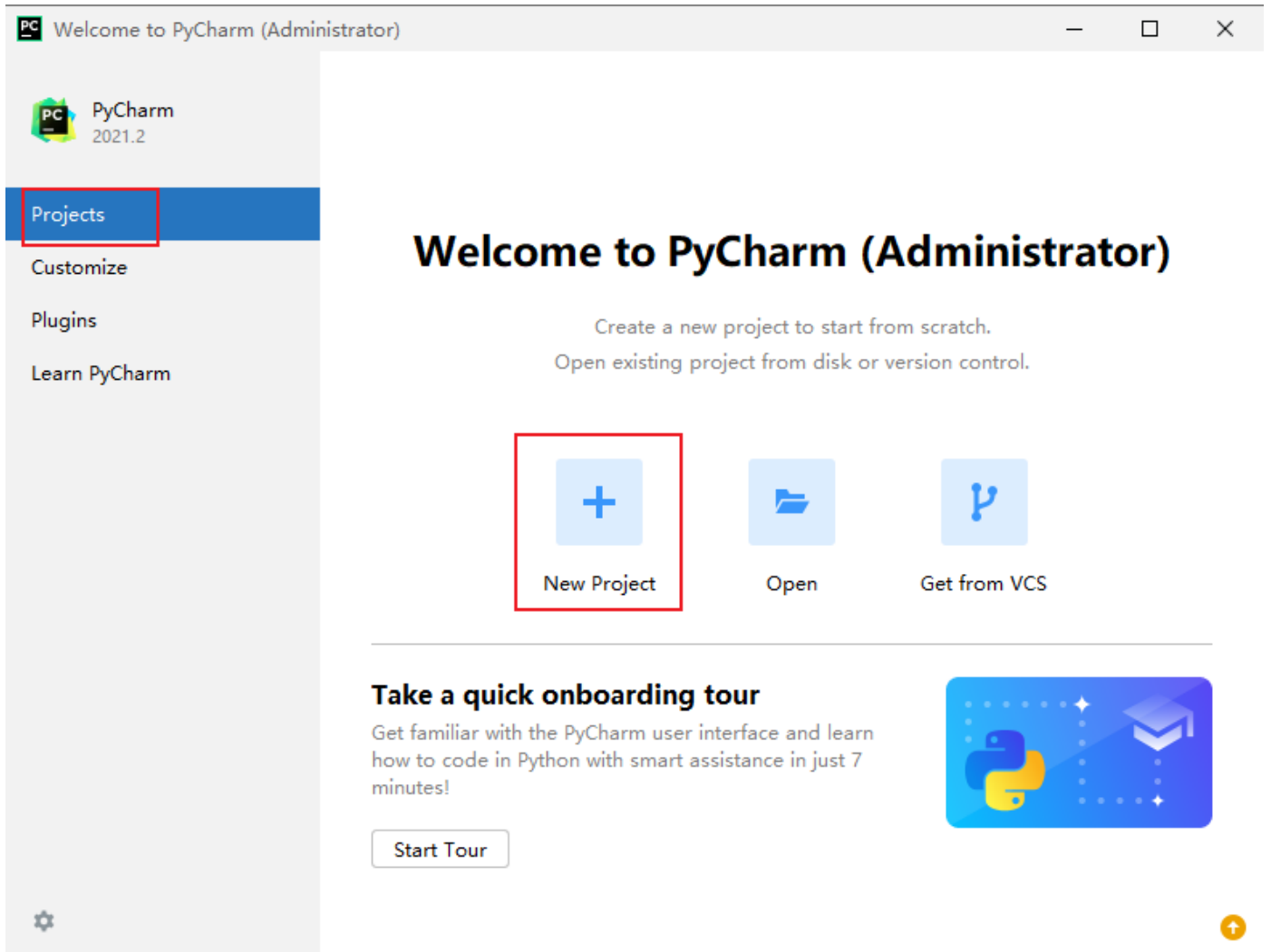
## 14、使用Python编写上传HDFS代码

说明：

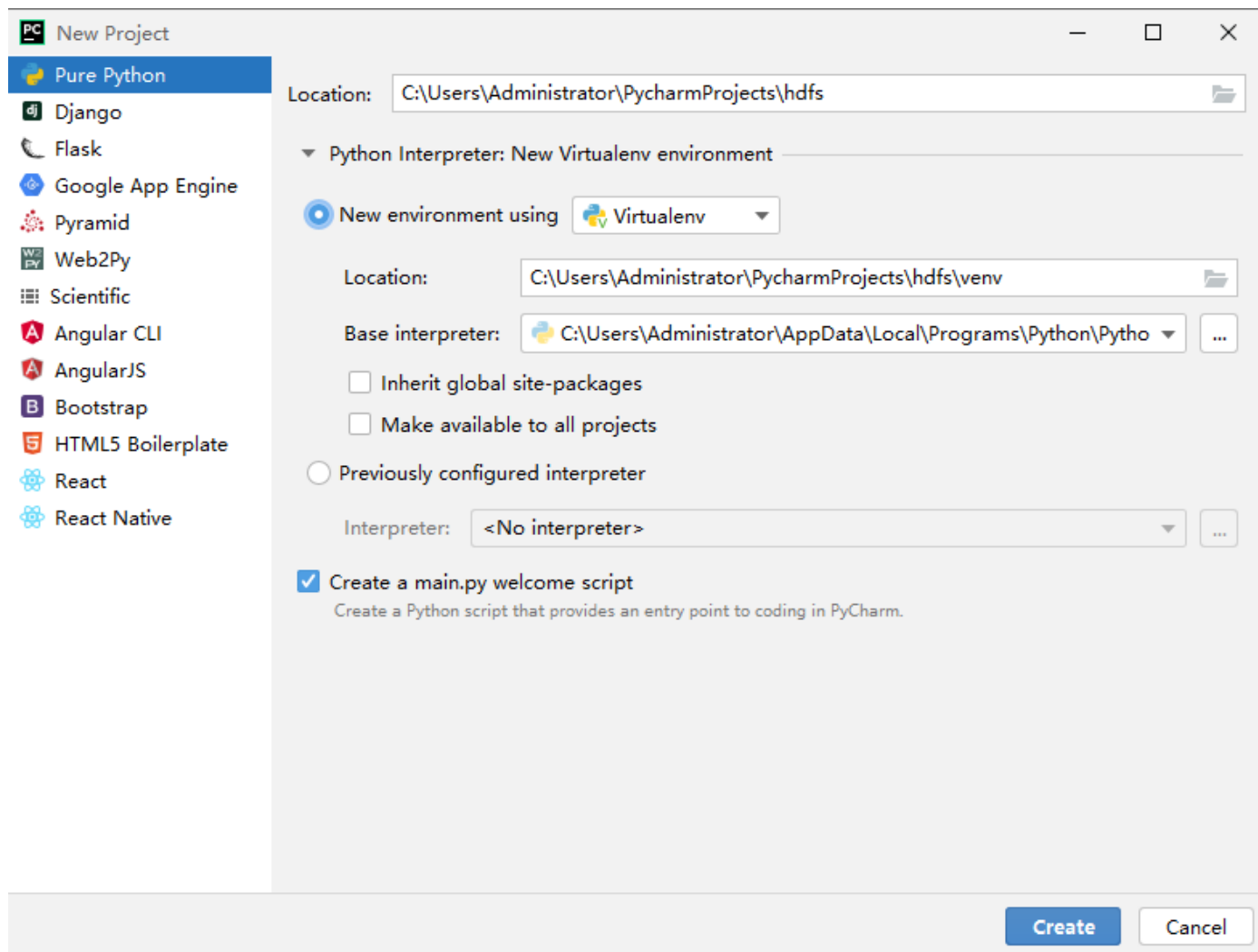
- 1、使用PyCharm新建upfile.py文件；
- 2、编写代码实现将本地日志文件D:\edits\logs\2022-03-09.log上传至HDFS服务器(192.168.100.128:9000)的效果；

### 14.1、使用Pycharm创建Python项目

#### 14.1.1、打开pycharm选择New Project

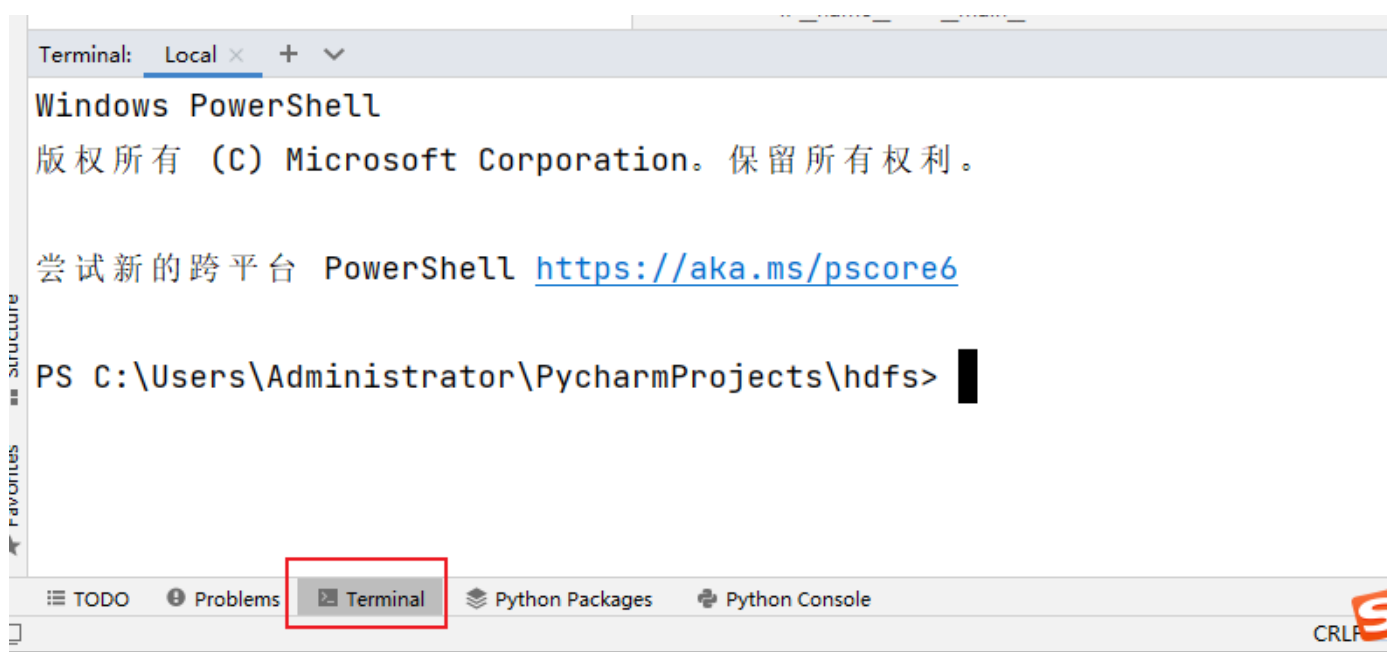


### 14.1.2、配置项目基本信息

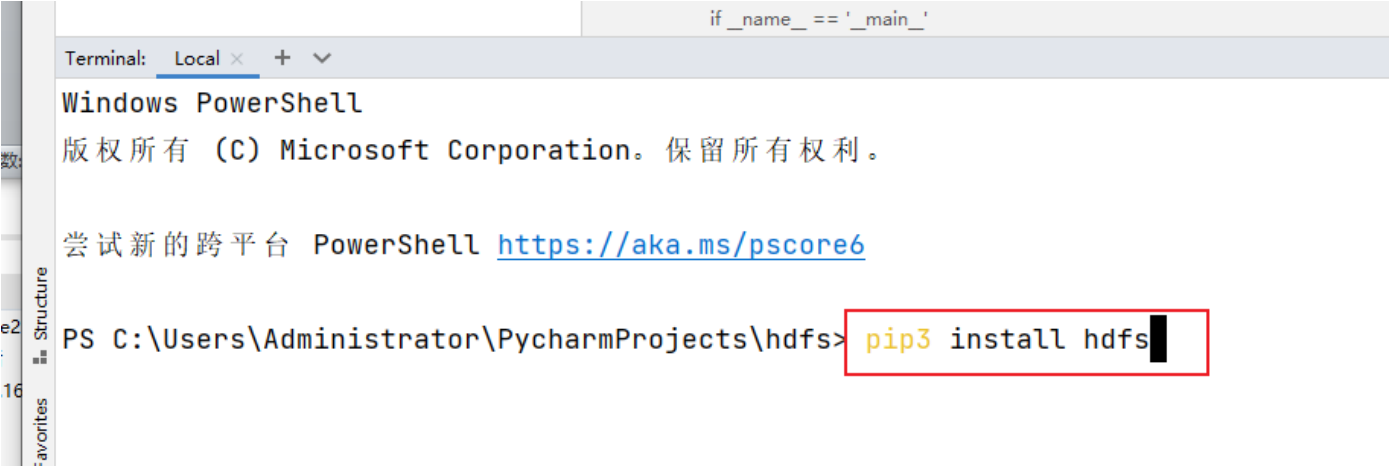


### 14.1.3、安装hdfs库

步骤一：点击pycharm底部工具栏中的Terminal（终端）

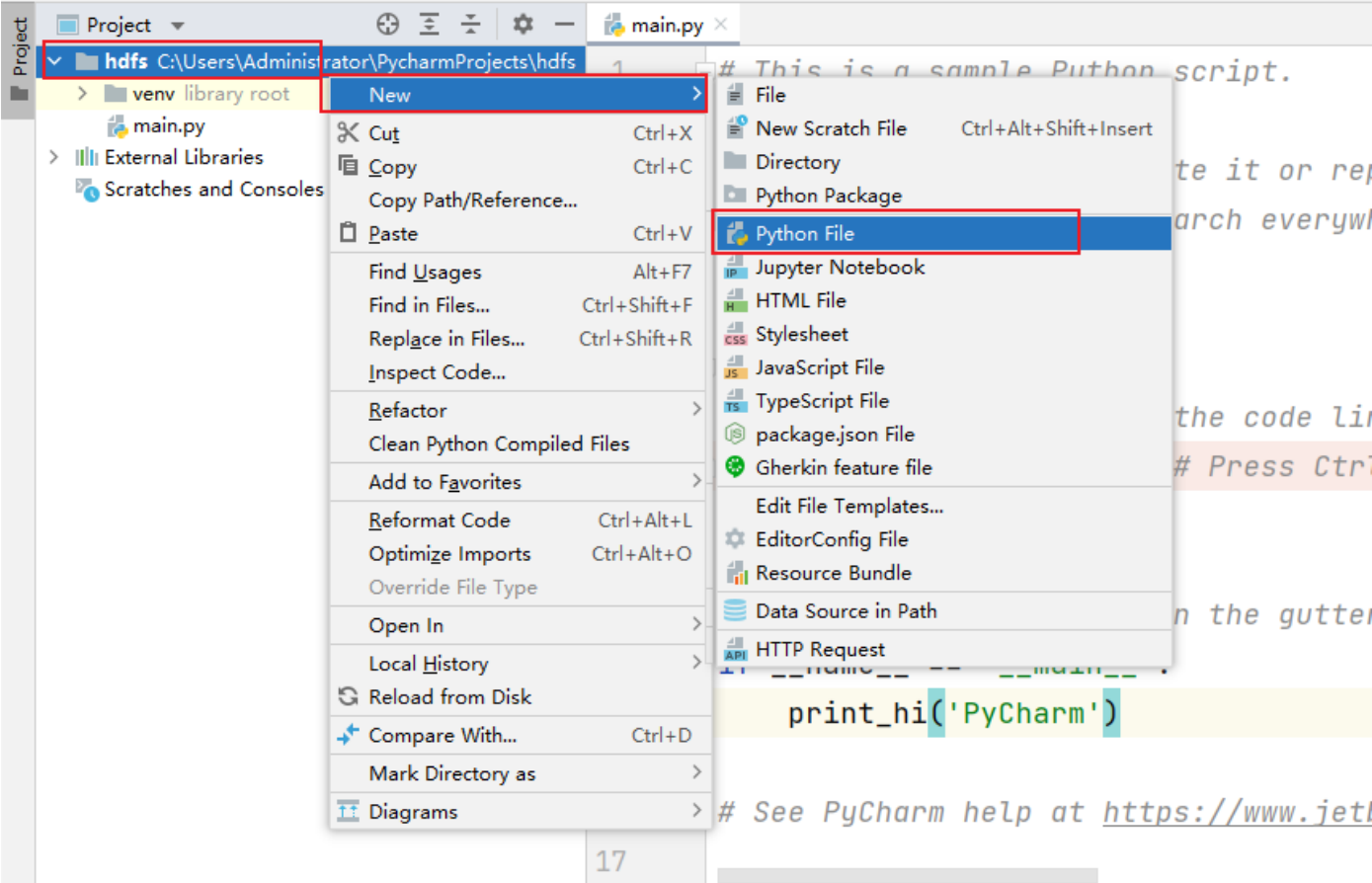


步骤二：在终端里输入 `pip3 install hdf5` 命令安装hdfs包



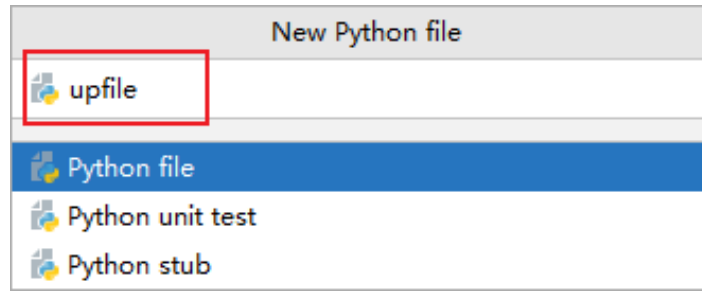
## 14.2、使用pycharm创建upfile.py文件

步骤一：鼠标点击项目名 --> 右键 --> New (新建) --> Python File



步骤二：在弹出的窗口中输入文件名：upfile





## 14.3、编写代码

步骤一：在upfile.py文件中编写代码

```
# 导入hdfs模块
from hdfs import *

# 创建HDFS客户端对象
cls = Client("http://192.168.100.128:50070", root='/', timeout=100, session=False)

# 异常处理
try:
    # 调用HDFS上传方法
    cls.upload('/', 'D:\\\\edits\\\\logs\\\\2022-03-09.log')
    print('文件上传成功')
except:
    print('文件上传失败')
```