1. **配置静态IP地址**
2. 在配置静态网络时先打开VMware里的编辑查看子网、子网掩码、网关、起始IP地址、结束IP地址
3. 打开 /etc/sysconfig/network-scripts/ifcfg-ens33

改变以下配置

<1>ONBOOT=no改为yes

<2>BOOTPROTO=dhcp改为static

<3>静态IP地址：IPADDR=  
<4>子网掩码：NETMASK=255.255.255.0

<5>网关地址：GATEWAY=

<6>与网关地址相同，域名解析的地址：DNS1=

更改完后重启网卡：systemctl restart network

1. 克隆节点

<1>克隆号一台虚拟机都要修改MAC地址  
<2>修改网络配置文件

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

<3>重启网卡:systemctl restart network

1. 修改真机IP

<1>打开win系统的网络设置，选择更改适配器选项

<2>打开VMnet8属性，选择Internet协议版本4

<3>点击使用下面的IP地址，

IP地址与前面的**2**中子网IP网段相同，最后一位是1；

子网掩码与2中相同；

默认网关与2中相同；

首先DNS服务器为8·8·8·8

1. 使用shell工具链接虚拟机
2. 编辑hosts文件：vim /etc/hosts

添加：192.168.174.140 node01

192.168.174.141 node02

192.168.174.142 node03

**7.**验证：ping node01

**8**.win与虚拟机ping通

路径：C/Windows/System32/dirvers/etc/hosts

Win+R验证

1. **实现三台机器免密登录**
2. 关闭防火墙和SELinux

<1>查看防火墙状态

systemctl status firewalld

<2>关闭防火墙

systemctl stop firewalld

<3>开机自动关闭防火墙

systemctl disable firewalld

<4>关闭SELinux

vim /etc/selinux/config

修改：SELINUX=disabled

重启查看状态：sestatus

1. 配置免密登录

<1>三台机器生成公钥和私钥

ssh-keygen -t rsa

<2>拷贝公钥到同一台机器上

ssh-copy-id node01

<3>复制第一台机器的认证到其它机器上面

scp -r /root/.ssh/authorized\_keys node02:/root/.ssh

<4>验证免密登录：ssh+主机名

1. **三台机器配置jdk**

**1.** 查看自带的jdk并卸载

rpm -qa | grep java

yum -y remove java-1.\*

**2.** 创建安装目录

mkdir -p /export/softwares #软件包存放目录

mkdir -p /export/servers #安装目录

**3.**上传并解压

tar -zxvf jdk-8u141-linux-x64.tar.gz -C ../servers/

**4.** 配置环境变量

vim /etc/profile

添加如下内容

export JAVA\_HOME=/export/servers/jdk1.8

export PATH=:$JAVA\_HOME/bin:$PATH

修改完成之后记得 source /etc/profile生效

**5.** 然后通过 scp指令传到2，3号机器上

scp -r /export/servers/jdk1.8/ node02:/export/servers

**6.**上传环境变量

scp -r /etc/profile node02:/etc/profile

**7.**统一生效环境变量文件

source /etc/profile

1. **安装mysql**

**1.** 先查看是否安装mariadb

rpm -qa | grep mariadb

**2.** 如果找到，则拷贝结果，使用下面命令删除

rpm -e --nodeps mariadb-libs-5.5.35-3.el7.x86\_64

**3.** 卸载先停掉mysql进程   没有安装过的可以直接跳过

pkill -9 mysqld

rpm -qa|grep -i mysql

用命令 yum -y remove 卸载

yum -y remove mysql-community-client-5.6.38-2.el7.x86\_64

卸载不掉的用 rpm -ev

**4.**上传mysql安装包

yum -y install mysql57-community-release-el7-10.noarch.rpm

yum -y install mysql-community-server

启动mysql服务

service mysqld restart

重置密码，先获取原始密码，如下

grep "password" /var/log/mysqld.log

mysql -u root -p   密码 进入

修改密码语句如：alter user 'root'@'localhost' identified by 'Root!!2018';

最后刷新权限 flush privileges;

1. **安装hadoop集群**

**1、**下载、上传、安装

**2、**修改7个文件

**<1> core-site.xml**

修改内容：

<configuration>

<!-- 指定集群的文件系统类型:分布式文件系统 -->

<property>

<name>fs.default.name</name>

<value>hdfs://node01:8020</value>

</property>

<!-- 指定临时文件存储目录 -->

<property>

<name>hadoop.tmp.dir</name>

<value>/export/servers/hadoop-2.7.5/hadoopDatas/tempDatas</value>

</property>

<!-- 缓冲区大小，实际工作中根据服务器性能动态调整 -->

<property>

<name>io.file.buffer.size</name>

<value>4096</value>

</property>

<!-- 开启hdfs的垃圾桶机制，删除掉的数据可以从垃圾桶中回收，单位分钟 -->

<property>

<name>fs.trash.interval</name>

<value>10080</value>

</property>

</configuration>

**<2> hdfs-site.xml**

修改内容：

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>node01:50090</value>

</property>

<!-- 指定namenode的访问地址和端口 -->

<property>

<name>dfs.namenode.http-address</name>

<value>node01:50070</value>

</property>

<!-- 指定namenode元数据的存放位置 -->

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///export/servers/hadoop-2.7.5/hadoopDatas/namenodeDatas,file:///export/servers/hadoop-2.7.5/hadoopDatas/namenodeDatas2</value>

</property>

<!-- 定义dataNode数据存储的节点位置，实际工作中，一般先确定磁盘的挂载目录，然后多个目录用，进行分割 -->

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///export/servers/hadoop-2.7.5/hadoopDatas/datanodeDatas,file:///export/servers/hadoop-2.7.5/hadoopDatas/datanodeDatas2</value>

</property>

<!-- 指定namenode日志文件的存放目录 -->

<property>

<name>dfs.namenode.edits.dir</name>

<value>file:///export/servers/hadoop-2.7.5/hadoopDatas/nn/edits</value>

</property>

<property>

<name>dfs.namenode.checkpoint.dir</name>

<value>file:///export/servers/hadoop-2.7.5/hadoopDatas/snn/name</value>

</property>

<property>

<name>dfs.namenode.checkpoint.edits.dir</name>

<value>file:///export/servers/hadoop-2.7.5/hadoopDatas/dfs/snn/edits</value>

</property>

<!-- 文件切片的副本个数-->

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<!-- 设置HDFS的文件权限-->

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

<!-- 设置一个文件切片的大小：128M-->

<property>

<name>dfs.blocksize</name>

<value>134217728</value>

</property>

**<3> hadoop-env.sh**

修改内容：

export JAVA\_HOME=/export/servers/jdk1.8

**<4> mapred-site.xml (cp mapred-site.xml.template mapred-site.xml)**

修改内容：

<configuration>

<!-- 指定分布式计算使用的框架是yarn -->

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<!-- 开启MapReduce小任务模式 -->

<property>

<name>mapreduce.job.ubertask.enable</name>

<value>true</value>

</property>

<!-- 设置历史任务的主机和端口 -->

<property>

<name>mapreduce.jobhistory.address</name>

<value>node01:10020</value>

</property>

<!-- 设置网页访问历史任务的主机和端口 -->

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node01:19888</value>

</property>

</configuration>

**<5> yarn-site.xml**

修改内容：

<configuration>

<!-- 配置yarn主节点的位置 -->

<property>

<name>yarn.resourcemanager.hostname</name>

<value>node01</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<!-- 开启日志聚合功能 -->

<property>

<name>yarn.log-aggregation-enable</name>

<value>true</value>

</property>

<!-- 设置聚合日志在hdfs上的保存时间 -->

<property>

<name>yarn.log-aggregation.retain-seconds</name>

<value>604800</value>

</property>

<!-- 设置yarn集群的内存分配方案 -->

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>20480</value>

</property>

<property>

<name>yarn.scheduler.minimum-allocation-mb</name>

<value>2048</value>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>2.1</value>

</property>

</configuration>

**<6> mapred-env.sh**

修改内容：

export JAVA\_HOME=/export/servers/jdk1.8

**<7> slaves**

修改内容：

node01

node02

node03

**3. 创建出来配置文件中没有的目录**

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/tempDatas

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/namenodeDatas

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/namenodeDatas2

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/datanodeDatas

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/datanodeDatas2

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/nn/edits

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/snn/name

mkdir -p /export/servers/hadoop-2.7.5/hadoopDatas/dfs/snn/edits

分发安装包：

cd /export/servers/

scp -r hadoop-2.7.5 node02:$PWD

scp -r hadoop-2.7.5 node03:$PWD

**4.配置环境变量**

vim /etc/profile

export HADOOP\_HOME=/export/servers/hadoop-2.7.5

export PATH=:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

scp -r /etc/profile node02:/etc/

scp -r /etc/profile node03:/etc/

三台机器统一执行source /etc/profile

**5.启动集群**

<1>第一次启动集群时需要对namenode节点进行格式化操作（以后再启动绝对不能再格式化）

cd /export/servers/hadoop-2.7.5

bin/hdfs namenode -format

<2>启动

cd /export/servers/hadoop-2.7.5/sbin

start-all.sh(只需要在node01上面执行)

mr-jobhistory-daemon.sh start historyserver 历史任务管理器

<3>三个端口查看界面

[http://node01:50070/explorer.html#/](#/) 查看hdfs

<http://node01:8088/cluster> 查看yarn集群

<http://node01:19888/jobhistory> 查看历史完成的任务

1. **安装Spark**

**1、上传解压安装**

**2、如果有权限问题，可修改为root**

chown -R root /export/servers/spark

chgrp -R root /export/servers/spark/

**3、修改配置文件`spark-env.sh`, 以指定运行参数**

cd /export/servers/spark/conf

cp spark-env.sh.template spark-env.sh

vi spark-env.sh

**3.1.1将以下内容复制进配置文件末尾**

# 指定 Java Home

export JAVA\_HOME=/export/servers/jdk1.8

# 指定 Spark Master 地址

export SPARK\_MASTER\_PORT=7077

3.1.2将以下内容复制到`spark-env.sh`的**末尾**, 配置 HistoryServer 启动参数, 使得 HistoryServer 在启动的时候读取 HDFS 中写入的 Spark 日志

# 指定 Spark History 运行参数

export SPARK\_HISTORY\_OPTS="-Dspark.history.ui.port=4000

-Dspark.history.retainedApplications=3

-Dspark.history.fs.logDirectory=hdfs://node01:8020/spark\_log"

# 指定 Spark 运行时参数

exportSPARK\_DAEMON\_JAVA\_OPTS="-Dspark.deploy.recoveryMode=ZOOKEEPER -Dspark.deploy.zookeeper.url=node01:2181,node02:2181,node03:2181

-Dspark.deploy.zookeeper.dir=/spark"

**3.2.1复制 spark-defaults.con**

cd /export/servers/spark/conf

cp spark-defaults.conf.template spark-defaults.conf

vi spark-defaults.conf

3.2.2将以下内容复制到`spark-defaults.conf`末尾处, 通过这段配置, 可以指定 Spark 将日志输入到 HDFS 中

spark.eventLog.enabled true

spark.eventLog.dir hdfs://node01:8020/spark\_log spark.eventLog.compress true

**4、修改配置文件 slaves, 以指定从节点为止, 从在使用 sbin/start-all.sh 启动集群的时候, 可以一键启动整个集群所有的 Worker**

**·**进入配置目录, 并复制一份新的配置文件, 以供在此基础之上进行修改

cd /export/servers/spark/conf

cp slaves.template slaves

vi slaves

**·**配置所有从节点的地址

node02

node03

**启动spark**

**Sbin/start-all.sh**

**进入spark**

**Bin/spark-shell –master local[2]**

**5、为 Spark 创建 HDFS 中的日志目录**

hdfs dfs -mkdir -p /spark\_log