目录

[Hadoop三大发行版本（了解） 3](#_Toc99557530)

[创建虚拟机 4](#_Toc99557531)

[安装虚拟机 5](#_Toc99557532)

[虚*拟机的克隆* 9](#_Toc99557533)

[配置网络 9](#_Toc99557534)

[修改克隆虚拟机的静态IP 12](#_Toc99557535)

[修改克隆机主机名 13](#_Toc99557536)

[使用xshell远程登陆 15](#_Toc99557537)

[hadoop集群搭建 17](#_Toc99557538)

[Java环境变量配置 18](#_Toc99557539)

[Hadoop环境变量配置 19](#_Toc99557540)

[Hadoop目录结构 22](#_Toc99557541)

[hadoop运行方式 25](#_Toc99557542)

[Hadoop运行模式 26](#_Toc99557543)

[完全分布式搭建 26](#_Toc99557544)

[scp实现安全拷贝 由于我们之前只是在以虚拟机上面配置好环境变量和安装包，我们需要在其他两台机器上分别安装hadoop和jdk，那么我们可以选择分发能更好的帮我们完成任务 27](#_Toc99557545)

[ssh免密登陆 32](#_Toc99557546)

[配置ssh 32](#_Toc99557547)

[检查是否安装ssh 32](#_Toc99557548)

[查看服务是否启动 32](#_Toc99557549)

[ssh免密登陆 33](#_Toc99557550)

[集群配置 36](#_Toc99557551)

[1）集群部署规划 36](#_Toc99557552)

[2）配置文件说明 37](#_Toc99557553)

[2） 配置集群 38](#_Toc99557554)

[将集群中的节点的配置文件分发到其他节点（当前是hadoop3） 41](#_Toc99557555)

[启动集群 46](#_Toc99557556)

[Jps: 50](#_Toc99557557)

## Hadoop三大发行版本（了解）

Hadoop三大发行版本：Apache、Cloudera、Hortonworks。

Apache版本最原始（最基础）的版本，对于入门学习最好。2006

Cloudera内部集成了很多大数据框架，对应产品CDH。2008

Hortonworks文档较好，对应产品HDP。2011

Hortonworks现在已经被Cloudera公司收购，推出新的品牌CDP。

**1）Apache Hadoop**

官网地址：http://hadoop.apache.org

下载地址：https://hadoop.apache.org/releases.html

**2）Cloudera Hadoop**

官网地址：https://www.cloudera.com/downloads/cdh

下载地址：https://docs.cloudera.com/documentation/enterprise/6/release-notes/topics/rg\_cdh\_6\_download.html

（1）2008年成立的Cloudera是最早将Hadoop商用的公司，为合作伙伴提供Hadoop的商用解决方案，主要是包括支持、咨询服务、培训。

**（2）2009年Hadoop的创始人Doug Cutting也加盟Cloudera公司**。Cloudera产品主要为CDH，Cloudera Manager，Cloudera Support

（3）CDH是Cloudera的Hadoop发行版，完全开源，比Apache Hadoop在兼容性，安全性，稳定性上有所增强。Cloudera的标价为每年每个节点**10000美元**。

（4）Cloudera Manager是集群的软件分发及管理监控平台，可以在几个小时内部署好一个Hadoop集群，并对集群的节点及服务进行实时监控。

**3）Hortonworks Hadoop**

官网地址：https://hortonworks.com/products/data-center/hdp/

下载地址：https://hortonworks.com/downloads/#data-platform

（1）2011年成立的Hortonworks是雅虎与硅谷风投公司Benchmark Capital合资组建。

**（2）公司成立之初就吸纳了大约25名至30名专门研究Hadoop的雅虎工程师，上述工程师均在2005年开始协助雅虎开发Hadoop，贡献了Hadoop80%的代码。**

（3）Hortonworks的主打产品是Hortonworks Data Platform（HDP），也同样是100%开源的产品，HDP除常见的项目外还包括了**Ambari**，一款开源的安装和管理系统。

（4）2018年Hortonworks目前**已经被Cloudera公司收购**。

# 创建虚拟机

1. **创建虚拟机**

创建虚拟机->自定义类型->稍后->Linux->centos 64位->起名hadoop01

默认：处理器核数为2，内存至少2G,磁盘容量最少40G

->选择NAT网络转换->i其他设备选择推荐-完成

1. **初始化虚拟机安装系统**

编辑->CD/DVD->选择镜像文件—>确定

开启虚拟机->根据屏幕提示选择安装（安装英文版，确认时区为上海东八时区，默认登录账号：root密码：1234

## 安装虚拟机

**安装模板虚拟机，IP地址92.168.121.134、主机名称**hadoop01**、内存**4G**、**硬盘50G

 [root@hadoop01 ~]# ifconfig  
 ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
        inet 192.168.121.134 netmask 255.255.255.0 broadcast 192.168.121.255  
        inet6 fe80::8d35:334c:37f:ba0a prefixlen 64 scopeid 0x20<link>  
        inet6 fe80::3d28:be0:46a6:60fb prefixlen 64 scopeid 0x20<link>  
        ether 00:0c:29:70:32:2b txqueuelen 1000 (Ethernet)  
        RX packets 3036 bytes 288203 (281.4 KiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 419 bytes 36667 (35.8 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 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  
 ​  
 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:c4:02:a4 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  
 ​

**hadoop01虚拟机配置要求如下（本文Linux系统全部以CentOS-7.5-x86-1804为例）**

**（1）使用yum安装需要虚拟机可以正常上网，yum安装前可以先测试下虚拟机联网情况**

[root@hadoop01 ~]# ping www.baidu.com

PING www.baidu.com (14.215.177.39) 56(84) bytes of data.

64 bytes from 14.215.177.39 (14.215.177.39): icmp\_seq=1 ttl=128 time=8.60 ms

64 bytes from 14.215.177.39 (14.215.177.39): icmp\_seq=2 ttl=128 time=7.72 ms

**（2）安装epel-release**

注：Extra Packages for Enterprise Linux是为“红帽系”的操作系统提供额外的软件包，适用于RHEL、CentOS和Scientific Linux。相当于是一个软件仓库，大多数rpm包在官方 repository 中是找不到的）

[root@hadoop01 ~]# yum install -y epel-release

3）注意：如果Linux安装的是最小系统版，还需要安装如下工具；如果安装的是Linux桌面标准版，不需要执行如下操作

* net-tool：工具包集合，包含ifconfig等命令

[root@hadoop01 ~]# yum install -y net-tools

* vim：编辑器

[root@hadoop01 ~]# yum install -y vim

**2）关闭防火墙，关闭防火墙开机自启**

[root@hadoop01 ~]# systemctl stop firewalld

[root@hadoop01 ~]# systemctl disable firewalld.service

注意：在企业开发时，通常单个服务器的防火墙时关闭的。公司整体对外会设置非常安全的防火墙

**3）创建root用户，并修改root用户的密码**

[root@hadoop01 ~]# useradd root

[root@hadoop01 ~]# passwd root

**4）配置root用户具有root权限，方便后期加sudo执行root权限的命令**

[root@hadoop01 ~]# vim /etc/sudoers

修改/etc/sudoers文件，在%wheel这行下面添加一行，如下所示：

## Allow root to run any commands anywhere

root ALL=(ALL) ALL

## Allows people in group wheel to run all commands

%wheel ALL=(ALL) ALL

root ALL=(ALL) NOPASSWD:ALL

注意：root这一行不要直接放到root行下面，因为所有用户都属于wheel组，你先配置了root具有免密功能，但是程序执行到%wheel行时，该功能又被覆盖回需要密码。所以root要放到%wheel这行下面。

**5）在/opt目录下创建文件夹，并修改所属主和所属组**

（1）在/opt目录下创建module、software文件夹

[root@hadoop01 ~]# mkdir /export/servers

[root@hadoop01 ~]# mkdir /opt/software

（2）修改module、software文件夹的所有者和所属组均为root用户

[root@hadoop01 ~]# chown root:root /export/servers

[root@hadoop01 ~]# chown root:root /opt/software

（3）查看module、software文件夹的所有者和所属组

[root@hadoop01 ~]# cd /opt/

[root@hadoop01 opt]# ll

总用量 12

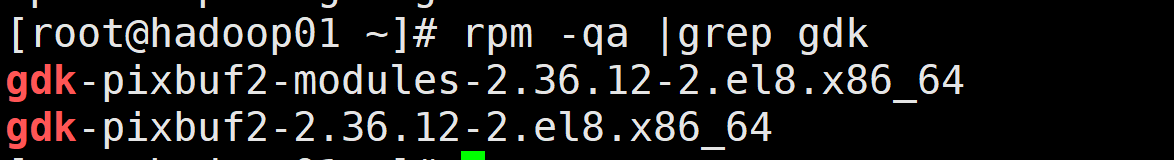
drwxr-xr-x. 2 root root 4096 5月 28 17:18 module

drwxr-xr-x. 2 root root 4096 9月 7 2017 rh

drwxr-xr-x. 2 root root 4096 5月 28 17:18 software

**6）卸载虚拟机自带的JDK**

**这个是因为我们通常在安装软件的时候会发现在使用rpm包，系统原本有rpm包，所以导致安装失败，所以需要卸载删除掉以前的rpm包**



注意：如果你的虚拟机是最小化安装不需要执行这一步。

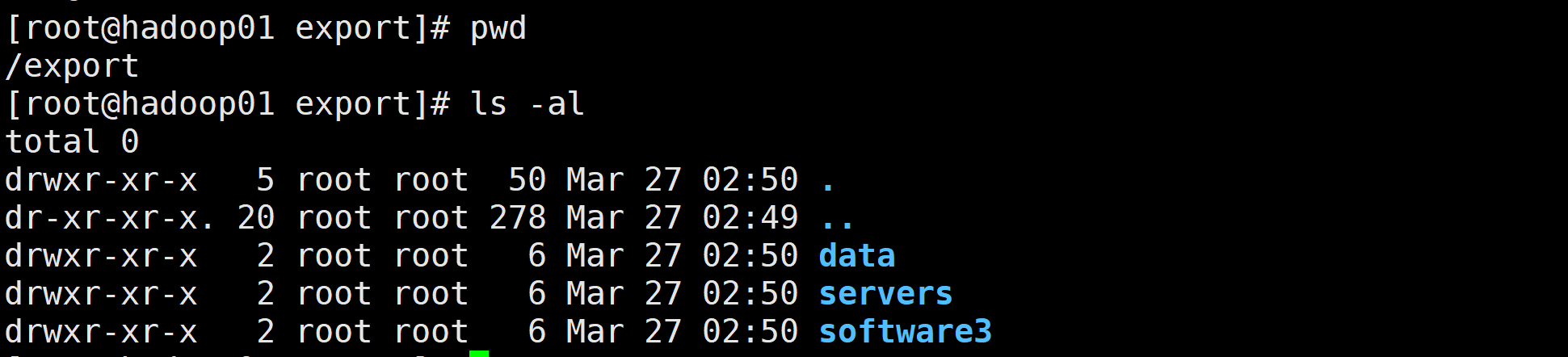
[root@hadoop01 ~]# rpm -qa | grep -i java | xargs -n1 rpm -e --nodeps

* rpm -qa：查询所安装的所有rpm软件包
* grep -i：忽略大小写
* xargs -n1：表示每次只传递一个参数
* rpm -e –nodeps：强制卸载软件

**7）重启虚拟机**

[root@hadoop01 ~]# reboot

**在本地Linux创建目录在本地目录下的/export/下分别创建data,servers,software3个目录，创建成功后查看**



# 虚拟机的克隆

1. 完整克隆： 对原始虚拟机完全独立的一个拷贝，它不和原始虚拟机共享任何资源，可以脱离原始虚拟机独立使用（适合用于开发）
2. 链接克隆：需要和原始虚拟机共享同一虚拟磁盘文件，不能脱离原始虚拟机独立运行，但是采用共享磁盘文件可以极大缩短创建克隆虚拟机的时间，同时还可以街上物理磁盘空间（用于个人测试）

**注意：克隆前需要关闭虚拟机**

**步骤：利用模板机hadoop100，克隆三台虚拟机：hadoop01 hadoop02 hadoop03**

**选中hadoop01->右键->管理->克隆->虚拟机当前状态-.完全克隆：名称为hadoop02**

**同样方法克隆hadoop03**

## 配置网络

**1. 编辑->虚拟网络编辑器->NAT模式->更改设置->NAT模式->改子网IP为**

**192.1683.121.0**

**2. NAT设置：网关为192.168.121.2**

**3. DHCP设置改为192.168.121.128-192.168.121.254**

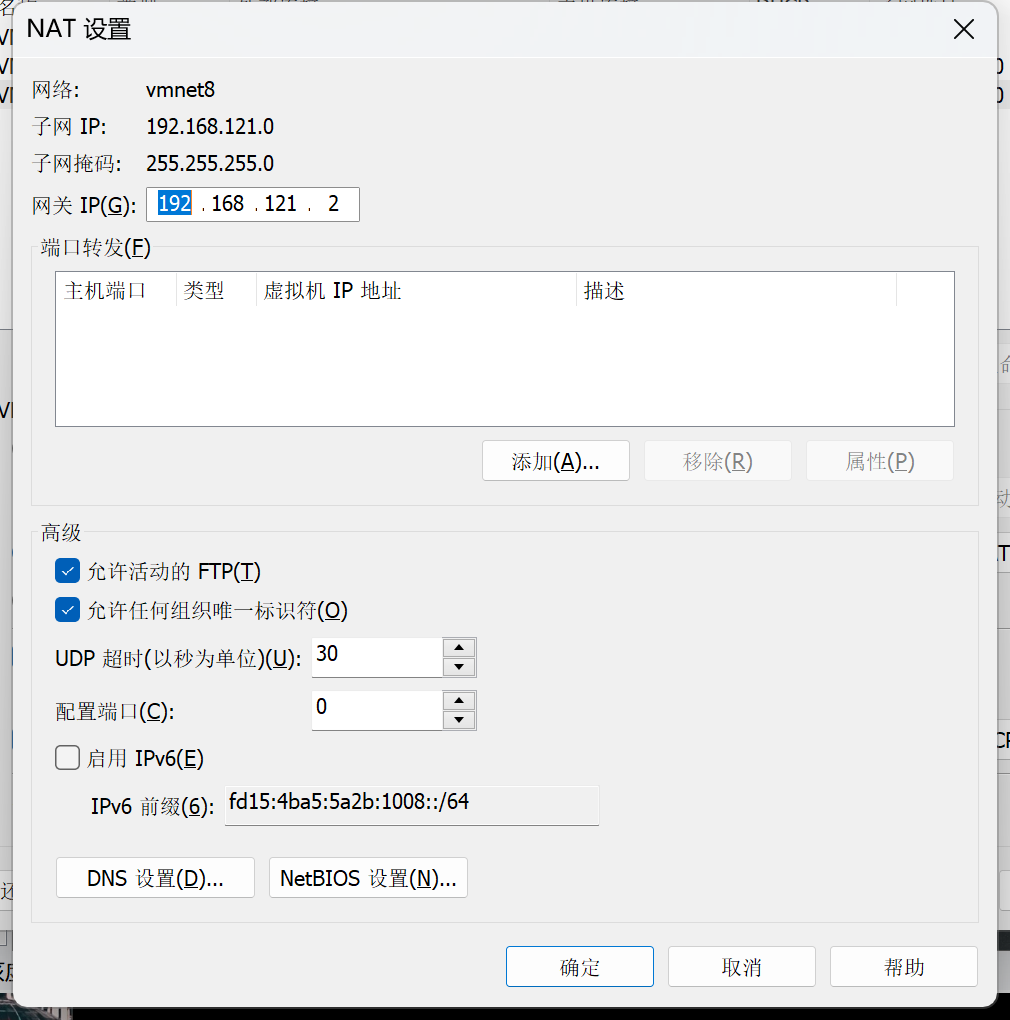
**4. 确定->应用->确定**

**5. 选择windows网络->更改适配器->vmware8->设置静态IP为192.168.121.5**

**6. 网关：192.168.121.2**

**7. DNS：8.8.8.8**







**注意：net模式下的子网和主机应该在同一个网段，或者会出现ping不同**

## 修改克隆虚拟机的静态IP

**vim /etc/sysconfig/network-scripts/ifcfg按tab自动补齐**

**一半路径是在vim /etc/sysconfig/network-scripts/ifcfg-ens33可能会出现Linux版本不同位置不同**

**保证Linux系统ifcfg-ens33文件中IP地址、虚拟网络编辑器地址和Windows系统VM8网络IP地址相同。**

**更改一下内容：**

 TYPE="Ethernet"  
 PROXY\_METHOD="none"  
 BROWSER\_ONLY="no"  
 BOOTPROTO="static"  
 DEFROUTE="yes"  
 IPV4\_FAILURE\_FATAL="no"  
 IPV6INIT="yes"  
 IPV6\_AUTOCONF="yes"  
 IPV6\_DEFROUTE="yes"  
 IPV6\_FAILURE\_FATAL="no"  
 IPV6\_ADDR\_GEN\_MODE="stable-privacy"  
 NAME="ens160"  
 UUID="4f0581bc-c00c-41c7-8fa3-fe3dda4c4433"  
 DEVICE="ens160"  
 ONBOOT="yes"  
 IPADDR=192.168.121.134  
 NETMASK=255.255.255.0  
 GATEWAY=192.168.121.2  
 DNS1=8.8.8.8

**改变** ：BOOTPROTO="static"

**添加：**

IPADDR=192.168.121.134  
 NETMASK=255.255.255.0  
 GATEWAY=192.168.121.2  
 DNS1=8.8.8.8

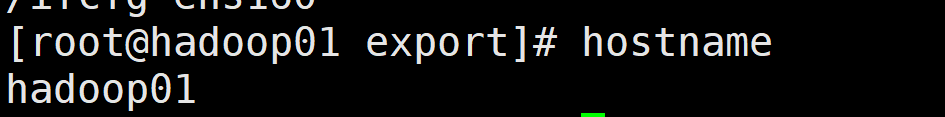
## 修改克隆机主机名

**主机名与ip地址映射配置(只改02和03）**

**配置主机名：**

vim /etc/sysconfig/network 更改hostname

**hostname:查看主机名称**



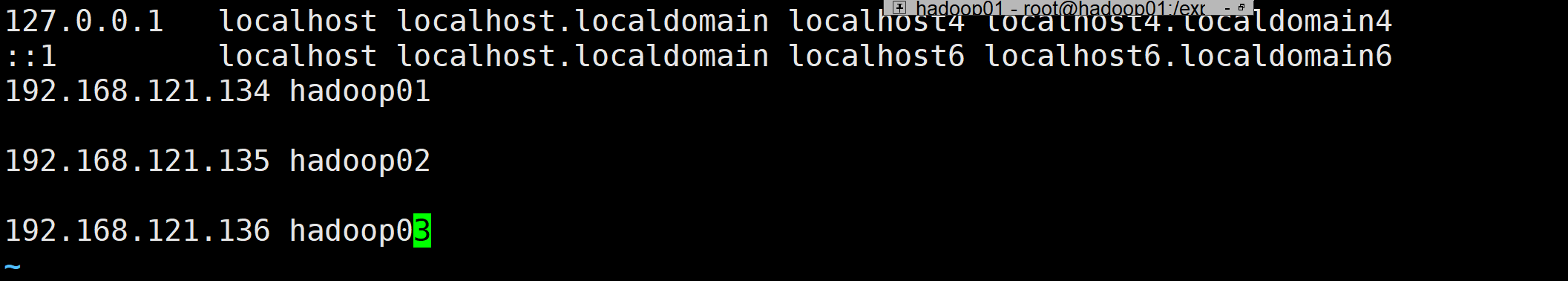
vim /etc/hostname

**查看ip地址可选范围（每台都要改）**

Vim /etc/hosts

**输入IP与主机名的映射**

192.168.121.134 hadoop01  
  
192.168.121.135 hadoop02  
  
192.168.121.136 hadoop03



**此时我们的其中一台虚拟机配置好了，再依次配置其他虚拟机**

**修改其他的虚拟机IP地址，输入IP与主机名的映射**

**，避免重复**

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

**依次测试网络并且关闭防火墙**

**测试hadoop03**

[root@hadoop03 ~]# hostname  
hadoop03  
[root@hadoop03 ~]# ping huya.com  
PING huya.com (106.52.191.105) 56(84) bytes of data.  
64 bytes from 106.52.191.105 (106.52.191.105): icmp\_seq=1 ttl=128 time=19.4 ms  
64 bytes from 106.52.191.105 (106.52.191.105): icmp\_seq=2 ttl=128 time=19.6 ms  
^C  
--- huya.com ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 3ms  
rtt min/avg/max/mdev = 19.381/19.492/19.604/0.178 ms

**测试hadoop02**

[root@hadoop02 ~]# hostname  
hadoop02  
[root@hadoop02 ~]# ping huya.com  
PING huya.com (106.52.191.105) 56(84) bytes of data.  
64 bytes from 106.52.191.105 (106.52.191.105): icmp\_seq=1 ttl=128 time=19.1 ms  
64 bytes from 106.52.191.105 (106.52.191.105): icmp\_seq=2 ttl=128 time=19.4 ms  
^C  
--- huya.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 11ms  
rtt min/avg/max/mdev = 18.742/19.417/20.117/0.490 ms

**测试hadoop01**

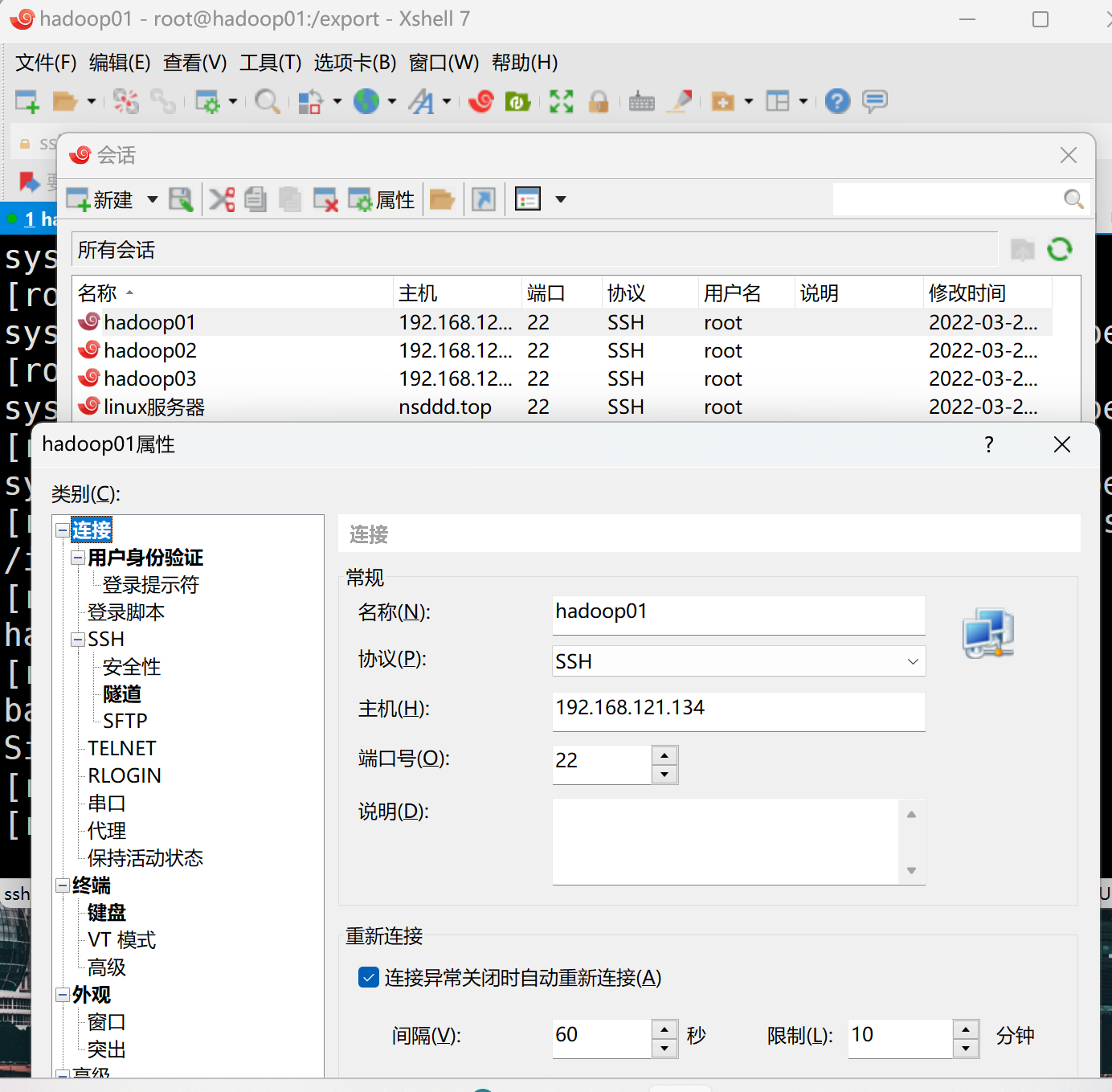
[root@hadoop01 ~]# hostname  
hadoop01  
[root@hadoop01 ~]# hostname  
hadoop01  
[root@hadoop01 ~]# ping huay.com  
PING huay.com (104.18.31.204) 56(84) bytes of data.  
64 bytes from 104.18.31.204 (104.18.31.204): icmp\_seq=1 ttl=128 time=152 ms  
^C  
--- huay.com ping statistics ---  
2 packets transmitted, 1 received, 50% packet loss, time 2ms  
rtt min/avg/max/mdev = 152.067/152.067/152.067/0.000 ms

**分别关闭防火墙**

systemctl stop firewalld  
systemctl disable firewalld.service

# 使用xshell远程登陆

使用xshell登陆是一个特别方便的选择，我们先下载好xshell



**方便windows登陆，我们可以修改windows下面的host文件**

**5）修改windows的主机映射文件（hosts文件）**

（1）如果操作系统是window7，可以直接修改

（a）进入C:\Windows\System32\drivers\etc路径

（b）打开hosts文件并添加如下内容，然后保存

192.168.121.134 hadoop01

192.168.121.135 hadoop02

192.168.121.136 hadoop03

（2）**如果操作系统是window10或者window11，先拷贝出来，修改保存以后，再覆盖即可**

（a）进入C:\Windows\System32\drivers\etc路径

（b）拷贝hosts文件到桌面

（c）打开桌面hosts文件并添加如下内容

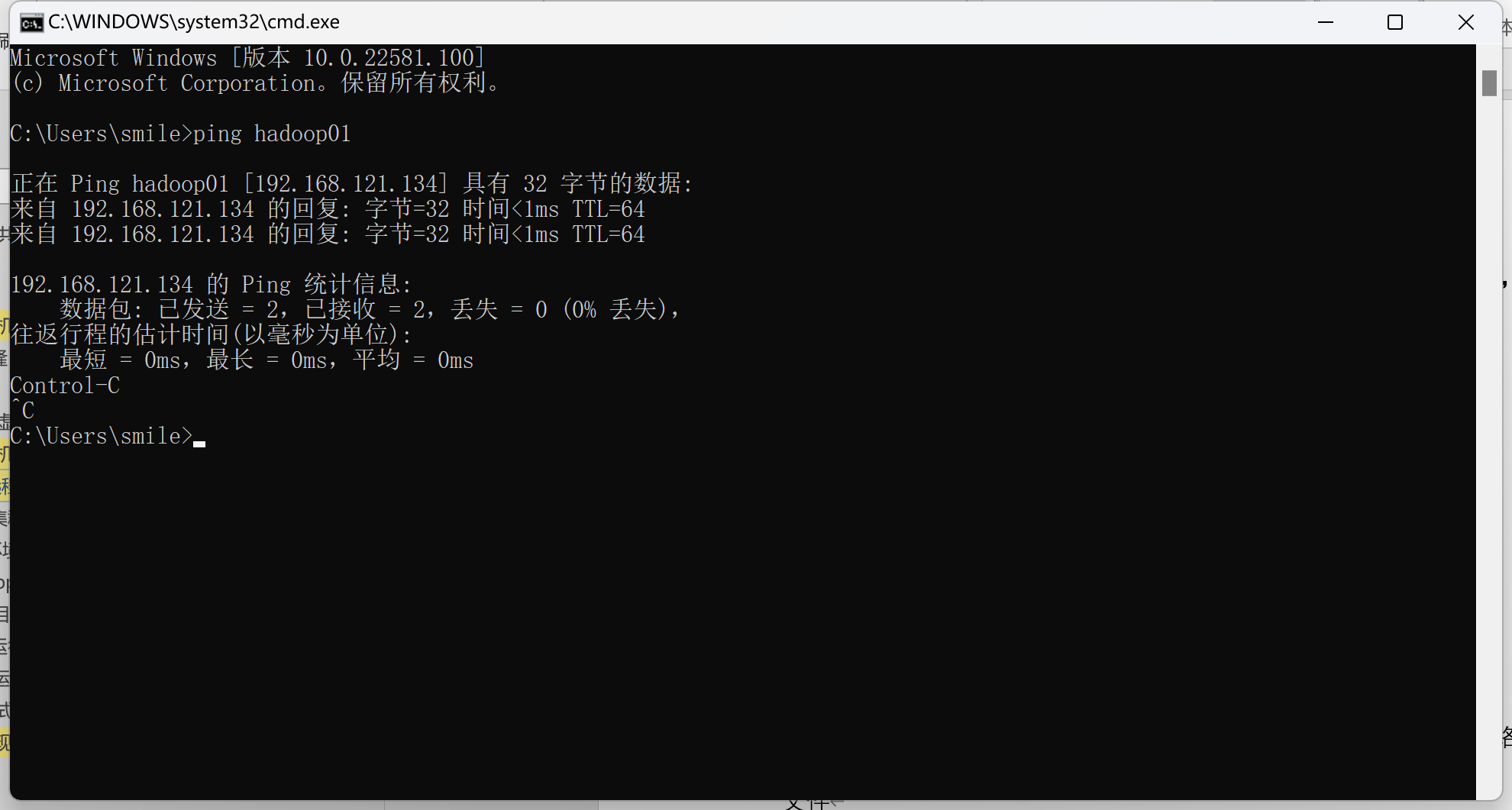
192.168.121.134 hadoop01

192.168.121.135 hadoop02

192.168.121.136 hadoop03

（d）将桌面hosts文件覆盖C:\Windows\System32\drivers\etc路径hosts文件

验证：

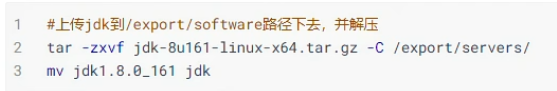


## hadoop集群搭建

**1）卸载现有JDK**

注意：安装JDK前，一定确保提前删除了虚拟机自带的JDK。

**2）用Xftp传输工具将JDK导入到opt目录下**



**3）配置环境变量**

### Java环境变量配置

**配置环境变量我们可以选择在/etc/profile中创建，但是我选择在profile.d中新建一个自定义的环境变量sh文件创建，方便管理**

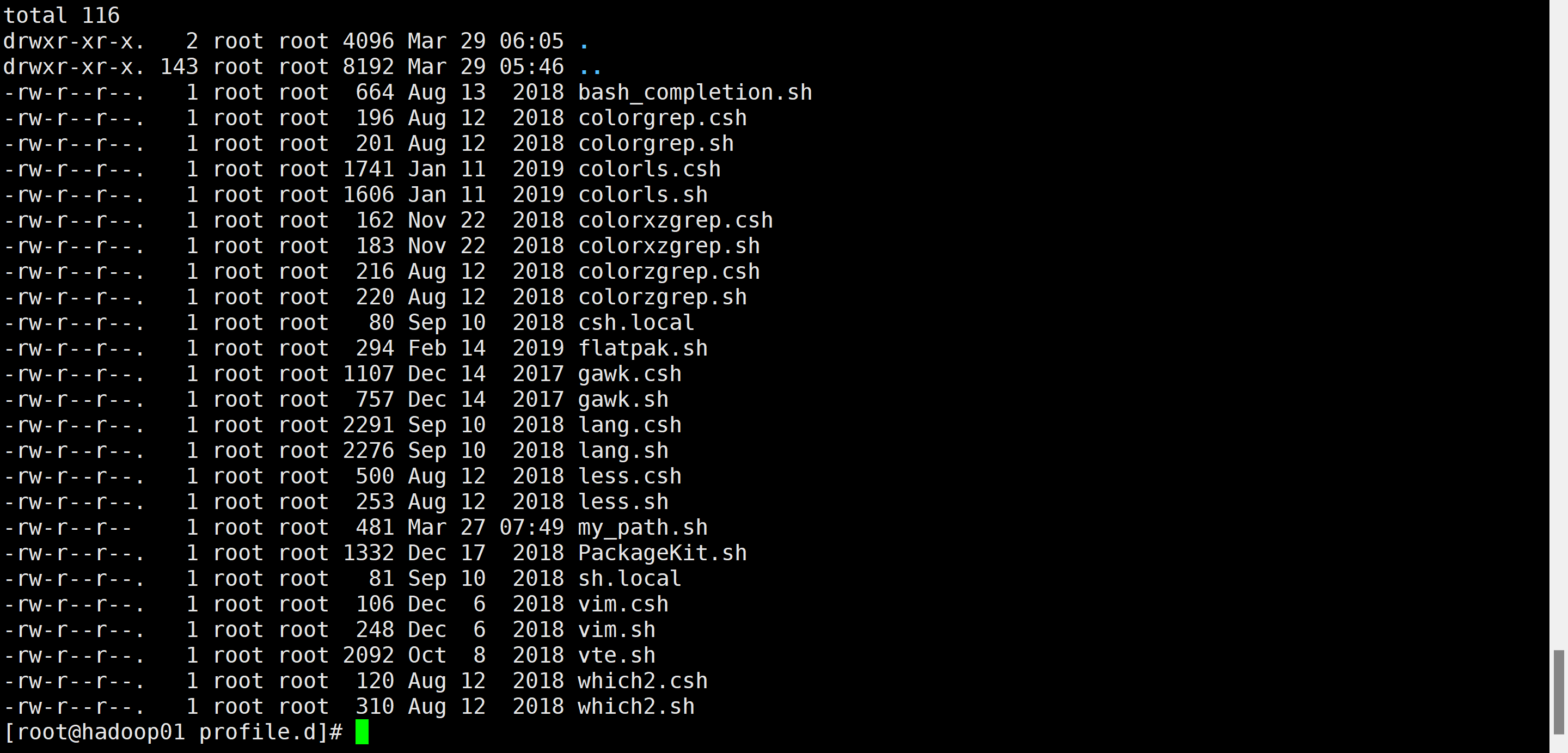
**我们新建一个自定义的类型，这个是基本的配置环境变量的目录**

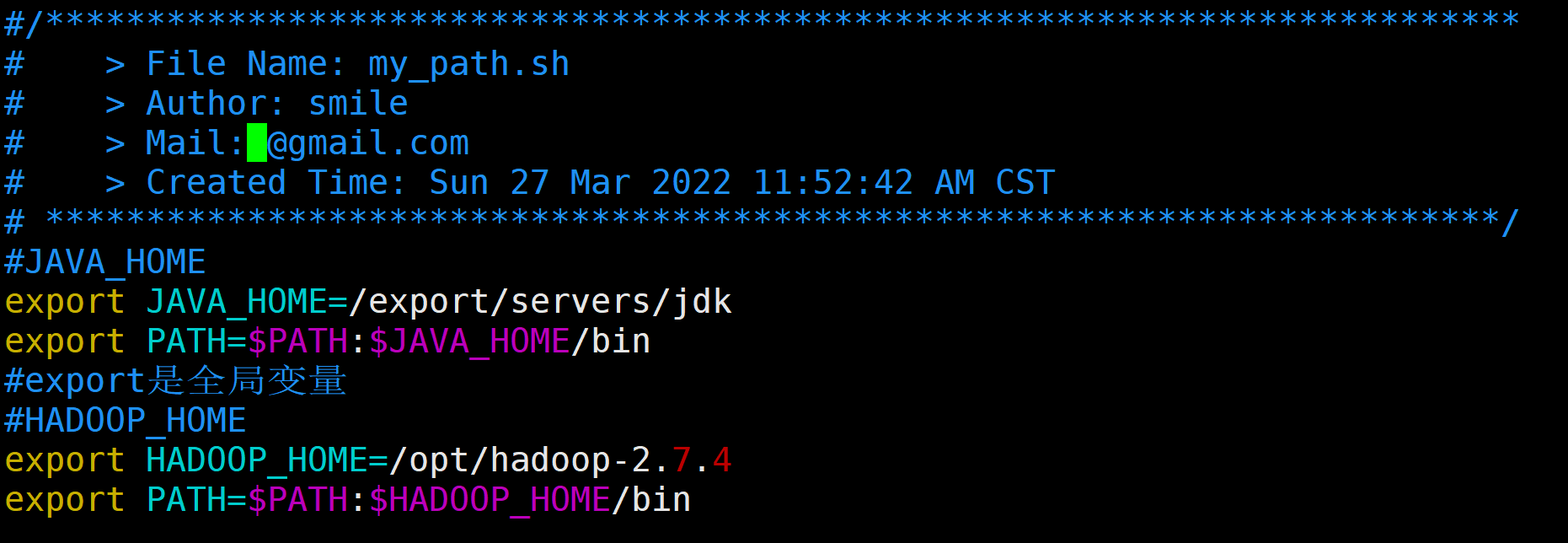
***配置环境变量目录***

~/.bashrc:该文件包含专用于某个用户的bash shell的bash信息,当该用户登录时以及每次打开新的shell时,该文件被读取.

/etc/profile中设定的变量(全局)的可以作用于任何用户,而~/.bashrc等中设定的变量(局部)只能继承/etc/profile中的变量,他们是"父子"关系

**在prpfile.d中创建一个文件，用来配置自己的环境变量，我命名为my\_path.sh**





**#JAVA\_HOME**

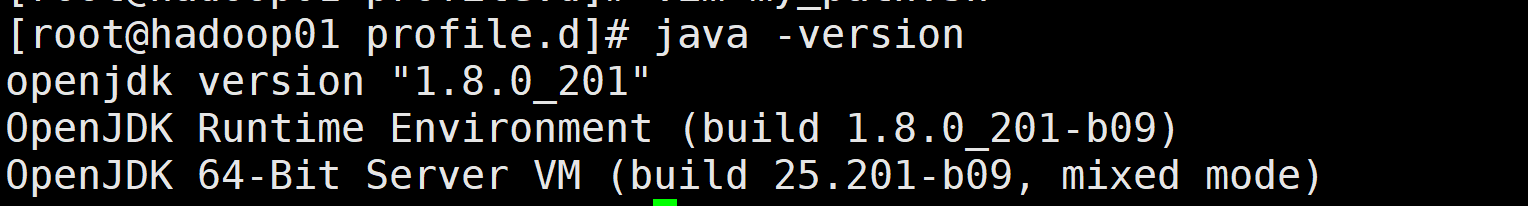
**export JAVA\_HOME=/export/servers/jdk**

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

在vim模式下wq退出

**Source /etc/profile使其生效，我们在profile代码中可以发现，shell也会遍历profile下面的文件，所以会更新profile.d下面的my\_path.sh**

**Java -version验证**



### Hadoop环境变量配置

类似于java的环境变量配置，我们同样将下载好的软件压缩包**解压在/export/servers/目录下面**

**解压命令：tar -zxvf 压缩包名称 压缩路径**

**在控制台：**

**[root@hadoop01 opt]# cd /etc/profile.d/**

**[root@hadoop01 profile.d]# vim my\_path.sh**

**进入vim按下G跳转到最后一排，然后按下o输入下面代码**

#HADOOP\_HOME

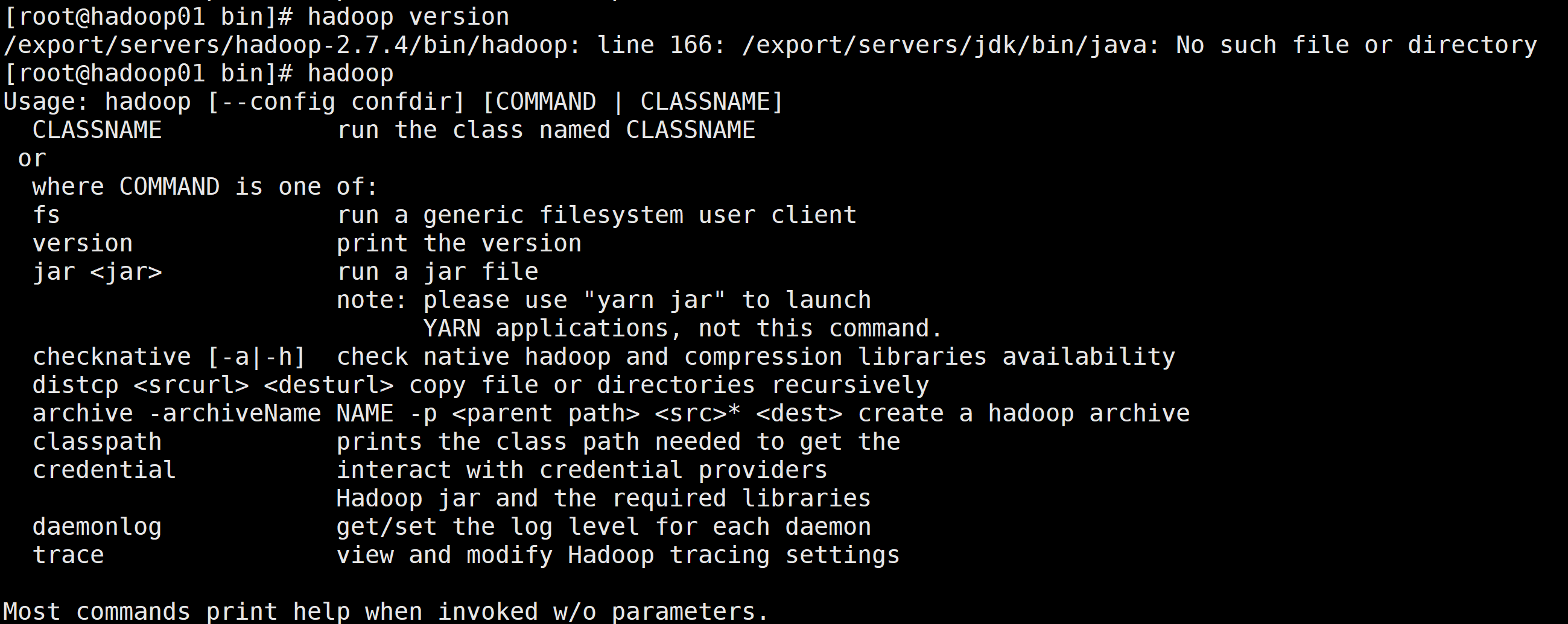
export HADOOP\_HOME=/export/servers/hadoop-3.3.2

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

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

**Source /etc/profile使其生效**

**验证Hadoop环境：  
Hadoop version**



**我们发现了hadoop version时候提醒我/export/servers/jdk/bin/java: No such file or directory，此时到文件里面查看166**



**无法解决???**

**which java : 查看Java指令位置**

**/usr/bin/java**

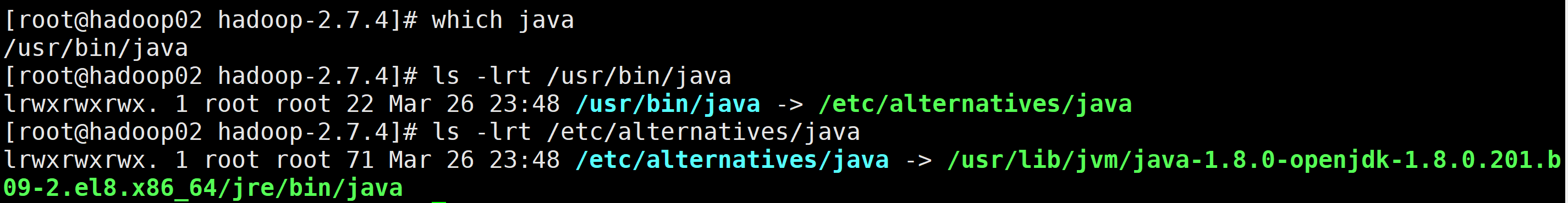
**再找到/usr/bin/java的超链接位置发现还是超链接**

**ls -lrt /usr/bin/java**

**lrwxrwxrwx 1 root root 22 Jul 27 11:43 /usr/bin/java -> /etc/alternatives/java**

**再来一次，发现最终位置**

**ls -lrt /etc/alternatives/java**

**lrwxrwxrwx. 1 root root 71 Mar 26 23:48 /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-2.el8.x86\_64/jre/bin/java**

**最后的这个jdk位置就是目前用的java的jdk位置**

**我们将这个存储在.bashrc里面 （vim ~/.bashrc）**

**输入G -> o在最末尾加上一句**

**export JAVA\_HOME=你的java安装路径**

**国内镜像下载gbk和hadoop**

**我发现系统提醒我们找不到jdk相关的头文件，我选择重新下载hadoop**

官网下载非常慢，找到了清华大学的镜像站下载很快  
清华大学镜像站：稳定版本的[https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/stable/](https://so.csdn.net/so/search?q=apache&spm=1001.2101.3001.7020" \t "_blank)

在Linux中可以直接使用命令：

**Wget https://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/stable/**

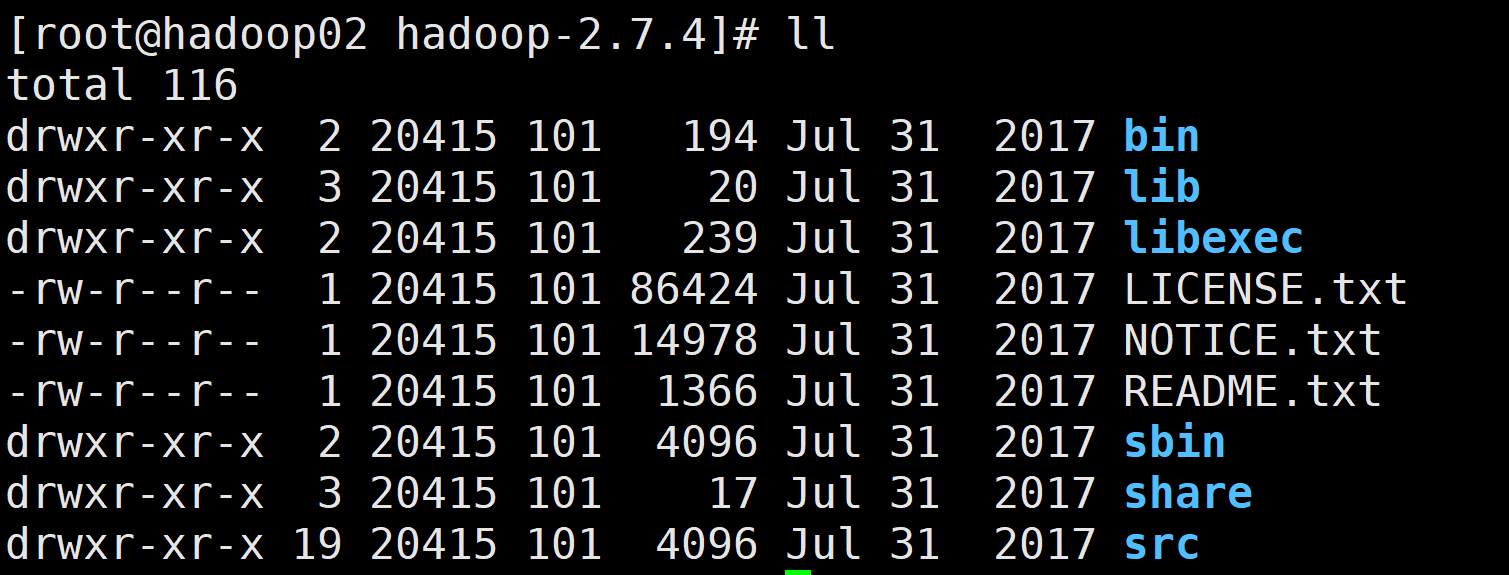
**Jbk:**

**Wget** [**https://repo.huaweicloud.com/java/jdk/12.0.2+10/jdk-12.0.2\_linux-x64\_bin.tar.gz**](https://repo.huaweicloud.com/java/jdk/12.0.2+10/jdk-12.0.2_linux-x64_bin.tar.gz)

**分别解压安装**

## Hadoop目录结构

1. **查看Hadoop目录结构**



**我发现这个版本的hadoop没有etc目录，所以建议直接使用wget下载新版的hadoop**

**使用新版地hadoop，配置环境变量：**

#/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# > File Name: my\_path.sh

# > Author: smile

# > Created Time: Sun 27 Mar 2022 11:52:42 AM CST

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#JAVA\_HOME

export JAVA\_HOME=/export/servers/jdk

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

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/too/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-2.el8.x86\_64/jre/bin/java

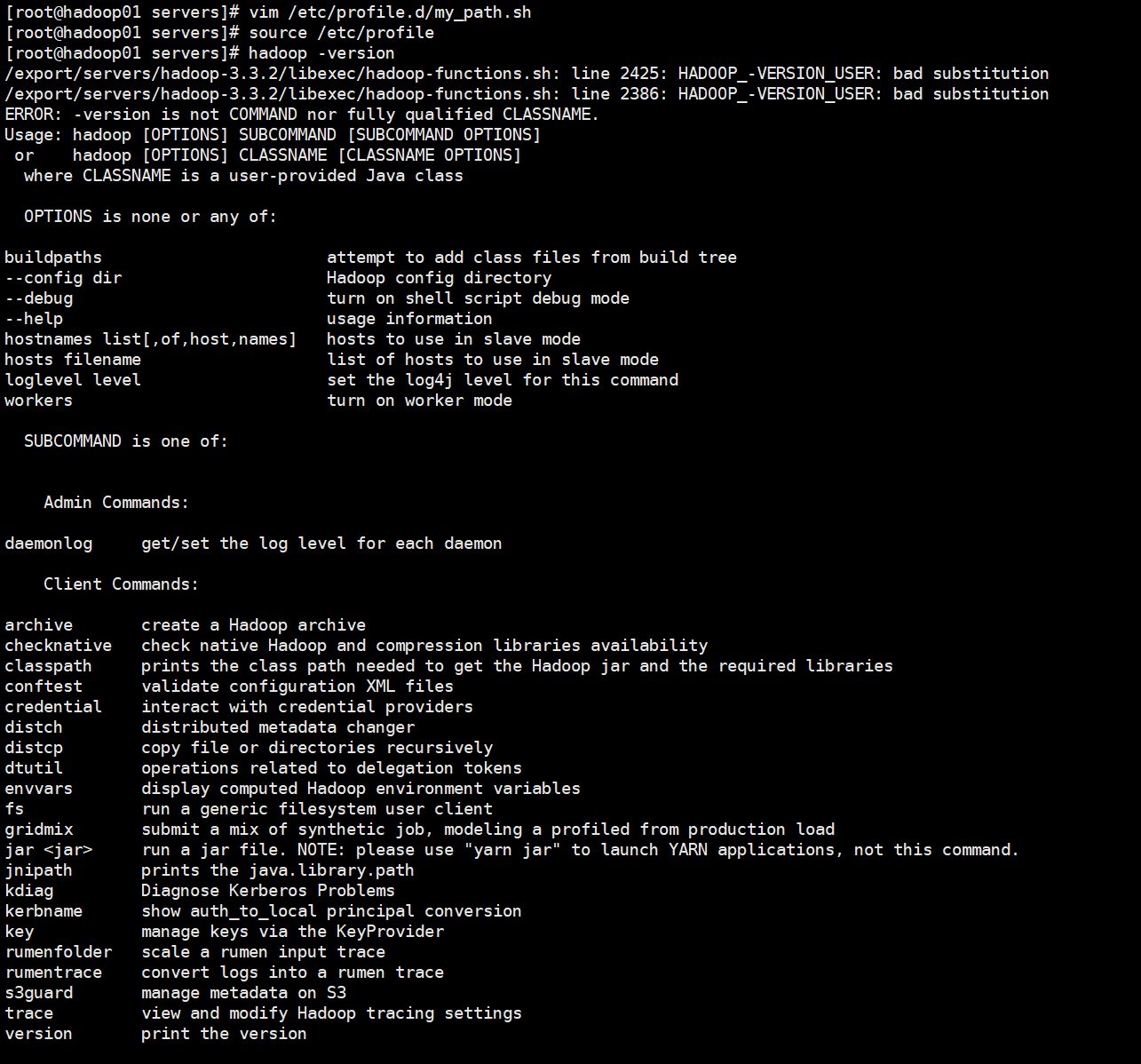
#HADOOP\_HOME

export HADOOP\_HOME=/export/servers/hadoop-3.3.2

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

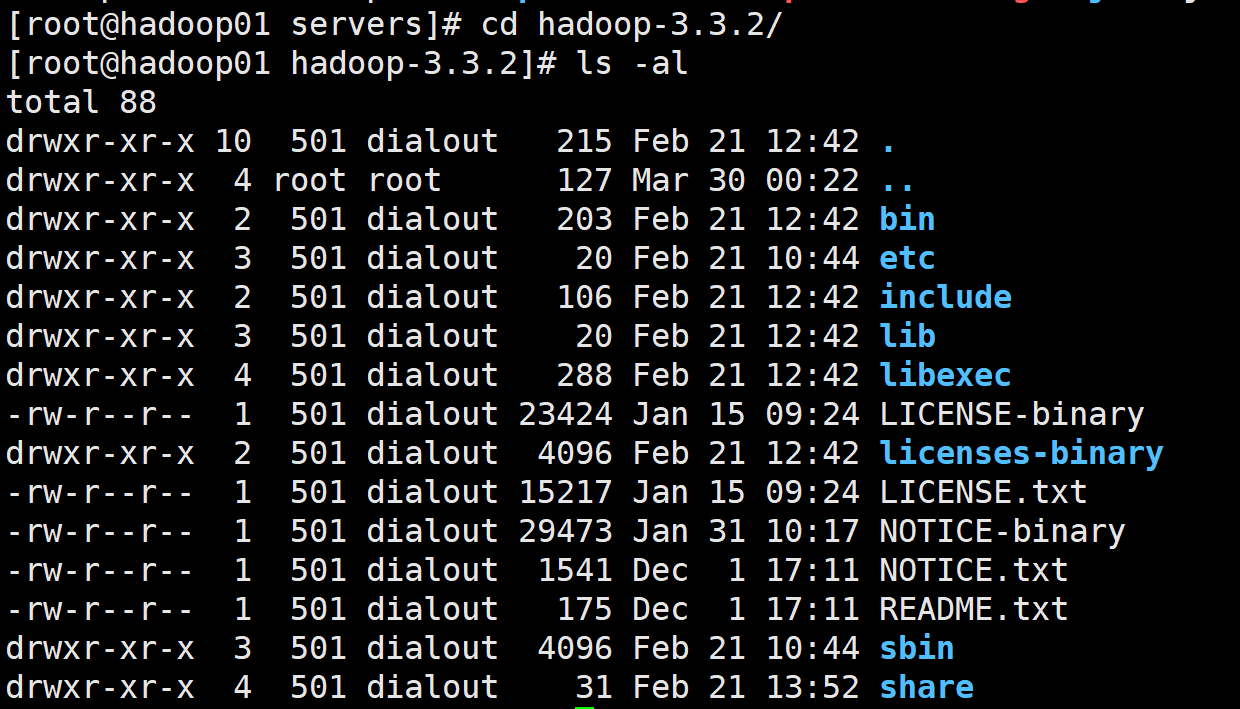
export PATH=$PATH:$HADOOP\_HOME/bin

**使用source /etc/profile刷新后使用hadoop -version：**



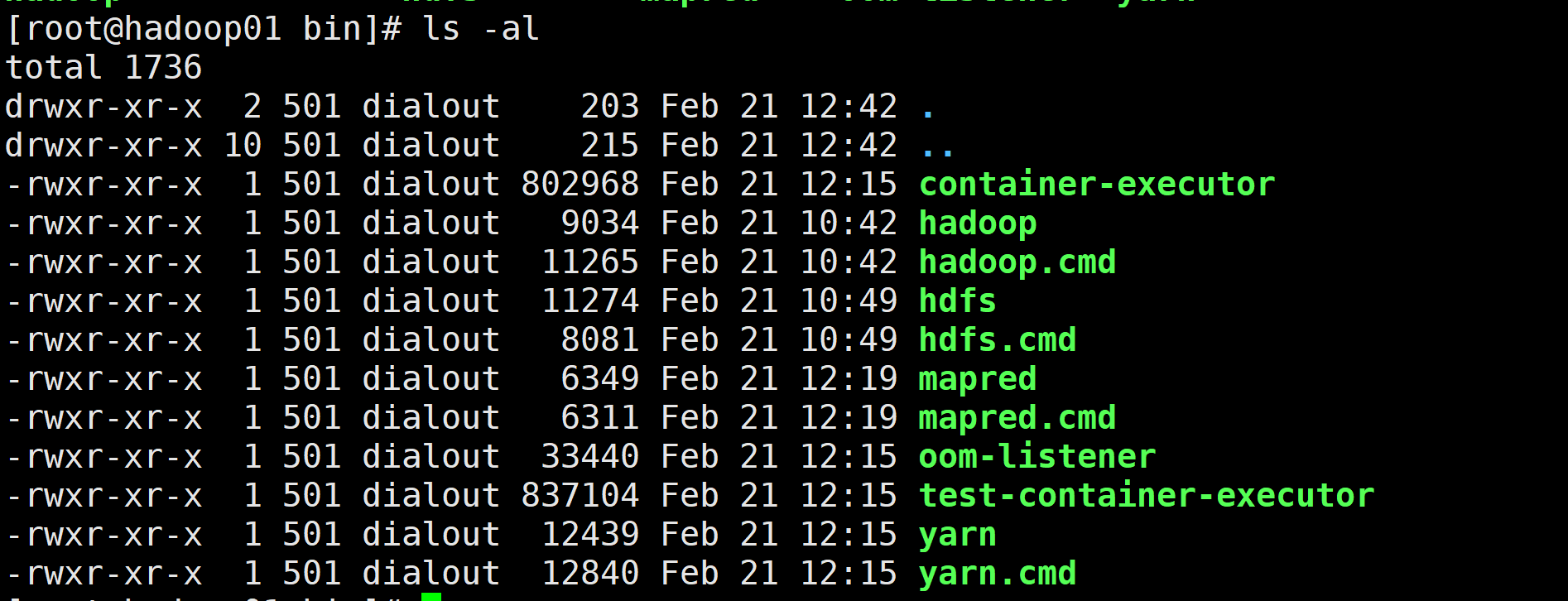
**有相关参数，说明配置成功！**

**新版的目录结构：**

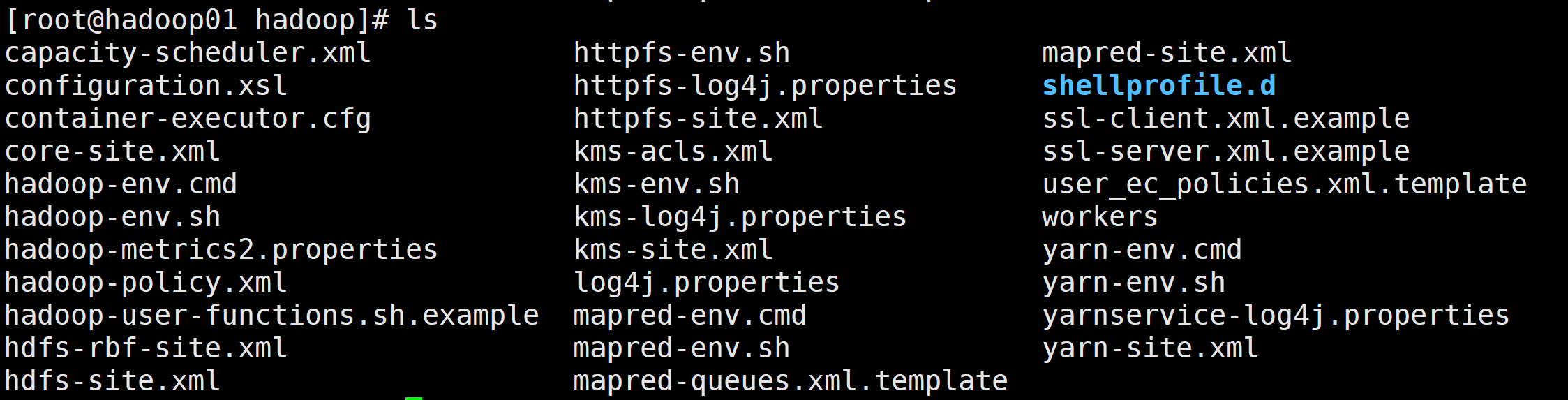


**2）重要目录**

（1）bin目录：存放对Hadoop相关服务（hdfs，yarn，mapred）进行操作的脚本

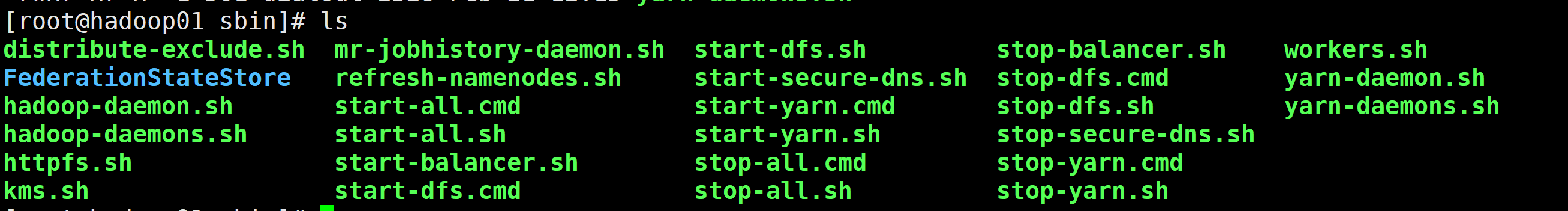


（2）etc目录：Hadoop的配置文件目录，存放Hadoop的配置文件



（3）lib目录：存放Hadoop的本地库（对数据进行压缩解压缩功能）

（4）sbin目录：存放启动或停止Hadoop相关服务的脚本

（5）share目录：存放Hadoop的依赖jar包、文档、和官方案例 

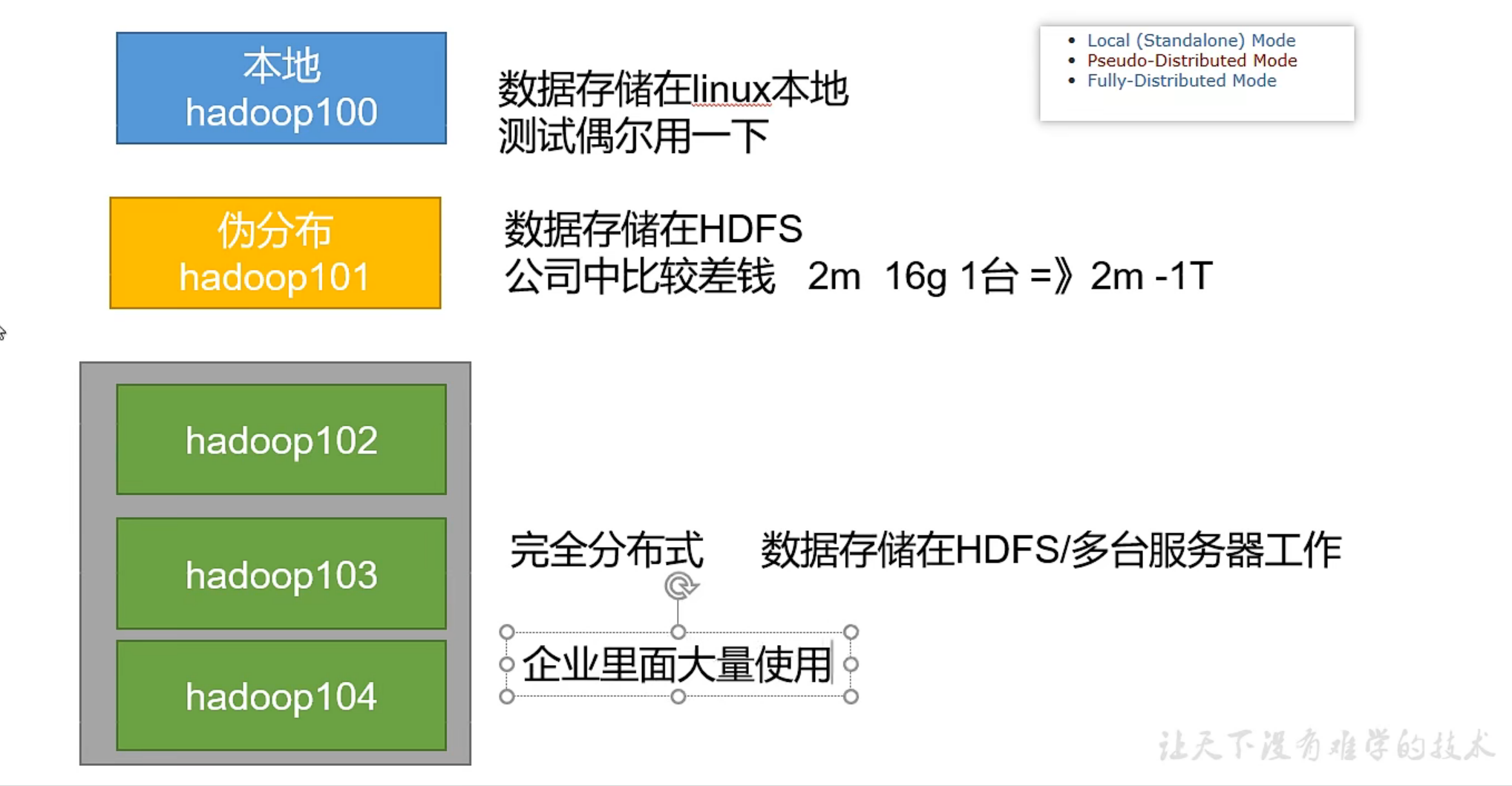
## hadoop运行方式

1）[Hadoop官方网站](http://hadoop.apache.org/)：<http://hadoop.apache.org/>

2）Hadoop运行模式包括：**本地模式**、**伪分布式模式**以及**完全分布式模式**。

* **本地模式**：单机运行，只是用来演示一下官方案例。生产环境不用。
* **伪分布式模式：**也是单机运行，但是具备Hadoop集群的所有功能，一台服务器模拟一个分布式的环境。个别公司用来测试，生产环境不用。
* **完全分布式模式：**多台服务器组成分布式环境。生产环境使用。

## Hadoop运行模式



## 完全分布式搭建

**分析步骤**

1. 安装三台虚拟机（关闭防火墙、静态ip、主机名称） （完成）
2. 安装jdk （完成）
3. 配置环境变量 （完成）
4. 安装hadoop （完成）
5. 配置环境变量 （完成）
6. 配置集群
7. 单点启动
8. 配置ssh (完成)
9. 群起并测试集群

**三台服务器有一台安装了jdk和hadoop此时我们需要编写一个脚本，将服务器中的程序拷贝**

### scp实现安全拷贝 由于我们之前只是在以虚拟机上面配置好环境变量和安装包，我们需要在其他两台机器上分别安装hadoop和jdk，那么我们可以选择分发能更好的帮我们完成任务

**1）scp（secure copy）安全拷贝**

（1）scp定义

scp可以实现服务器与服务器之间的数据拷贝。（from server1 to server2）

（2）基本语法

scp -r $pdir/$fname $user@$host:$pdir/$fname

命令 递归 要拷贝的文件路径/名称 目的地用户@主机:目的前提：在hadoop01、hadoop02、hadoop03都已经创建好的/xport/servers/ /opt/software两个目录，并且已经把这两个目录修改为root:root

[root@hadoop01 ~]#sudo chown root:root -R /xport/servers/

（a）在hadoop01上，将hadoop01中/xport/servers//jdk目录拷贝到hadoop02上。

[root@hadoop01 ~]#scp -r /xport/servers//jdk root@hadoop02:/xport/servers/

（b）在hadoop02上，将hadoop01中/xport/servers//hadoop-3.3.2目录拷贝到hadoop02上。

[root@hadoop02 ~]$ scp -r root@hadoop01:/xport/servers//hadoop-3.3.2 /xport/servers//

（c）在hadoop02上操作，将hadoop01中/xport/servers/目录下所有目录拷贝到hadoop03上。

[root@hadoop02 opt]$ scp -r root@hadoop01:/xport/servers//\* root@hadoop03:/xport/servers/

**2）rsync远程同步工具**

rsync主要用于备份和镜像。具有速度快、避免复制相同内容和支持符号链接的优点。

rsync和scp区别：用rsync做文件的复制要比scp的速度快，rsync只对差异文件做更新。scp是把所有文件都复制过去。

（1）基本语法

rsync -av $pdir/$fname $user@$host:$pdir/$fname

命令 选项参数 要拷贝的文件路径/名称 目的地用户@主机:目的地路径/名称

选项参数说明：

|  |  |
| --- | --- |
| 选项 | 功能 |
| -a | 归档拷贝 |
| -v | 显示复制过程 |

（a）删除hadoop02中/xport/servers//hadoop-3.3.2/wcinput

[root@hadoop02 hadoop-3.3.2]$ rm -rf wcinput/

（b）同步hadoop01中的/xport/servers//hadoop-3.3.2到hadoop02

[root@hadoop01 module]$ rsync -av hadoop-3.3.2/ root@hadoop02:/xport/servers//hadoop-3.3.2/

**3）xsync集群分发脚本**

（1）需求：循环复制文件到所有节点的相同目录下

（2）需求分析：

（a）rsync命令原始拷贝：

rsync -av /xport/servers/ root@hadoop02:/opt/

（b）期望脚本：

xsync要同步的文件名称

（c）期望脚本在任何路径都能使用（脚本放在声明了全局环境变量的路径）

[root@hadoop01 ~]#echo $PATH

/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/root/.local/bin:/home/root/bin:/xport/servers//jdk/bin

（3）脚本实现

（a）在/home/root/bin目录下创建xsync文件

[root@hadoop01 opt]$ cd /home/root

[root@hadoop01 ~]#mkdir bin

[root@hadoop01 ~]#cd bin

[root@hadoop01 bin]$ vim xsync

在该文件中编写如下代码

#!/bin/bash

#1. 判断参数个数

if [ $# -lt 1 ]

then

echo Not Enough Arguement!

exit;

fi

#2. 遍历集群所有机器

for host in hadoop01 hadoop02 hadoop03

do

echo ==================== $host ====================

#3. 遍历所有目录，挨个发送

for file in $@

do

#4. 判断文件是否存在

if [ -e $file ]

then

#5. 获取父目录

pdir=$(cd -P $(dirname $file); pwd)

#6. 获取当前文件的名称

fname=$(basename $file)

ssh $host "mkdir -p $pdir"

rsync -av $pdir/$fname $host:$pdir

else

echo $file does not exists!

fi

done

done

（b）修改脚本 xsync 具有执行权限

[root@hadoop01 bin]$ chmod +x xsync

（c）测试脚本

[root@hadoop01 ~]#xsync /home/root/bin

（d）将脚本复制到/bin中，以便全局调用

[root@hadoop01 bin]$ sudo cp xsync /bin/

（e）同步环境变量配置（root所有者）

[root@hadoop01 ~]#sudo ./bin/xsync /etc/profile.d/my\_env.sh

注意：如果用了sudo，那么xsync一定要给它的路径补全。

让环境变量生效

[root@hadoop02 bin]$ source /etc/profile

[root@hadoop03 opt]$ source /etc/profile

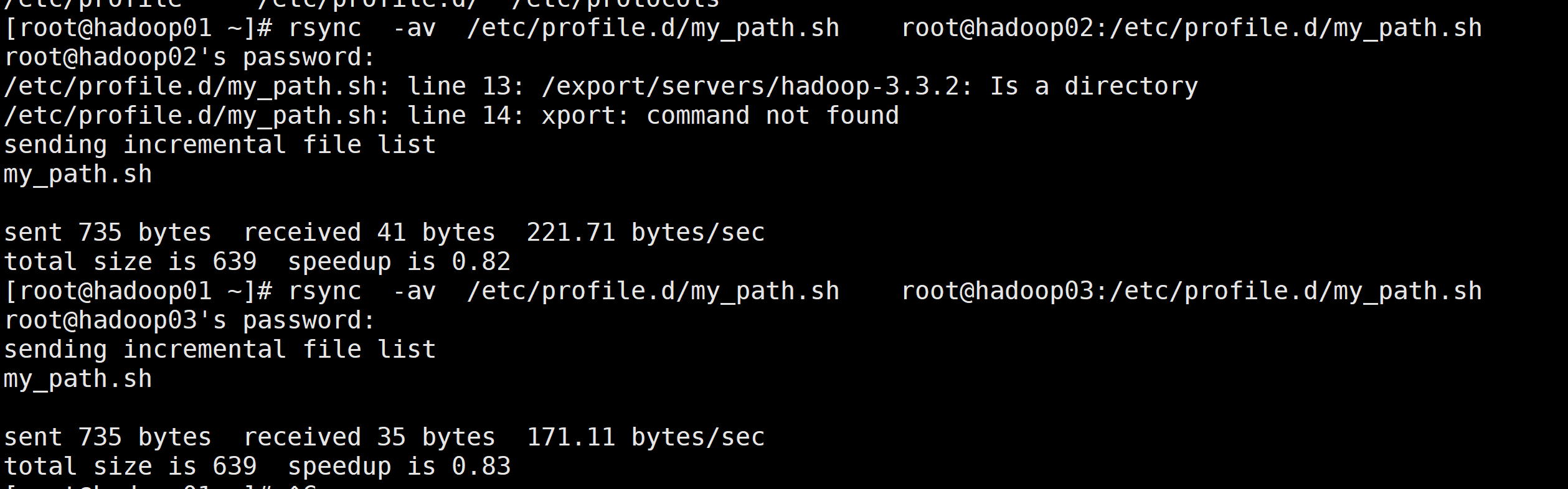
**或者使用下面方法修改配置文件：**

rsync -av /etc/profile.d/my\_path.sh root@hadoop03:/etc/profile.d/my\_path.sh

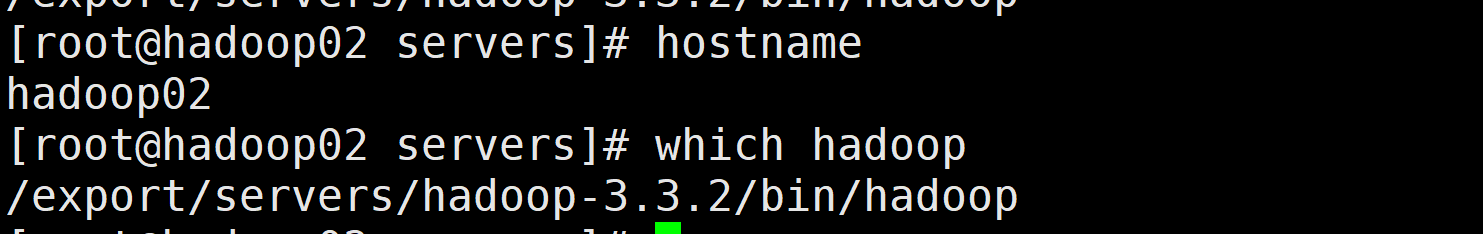
rsync -av /etc/profile.d/my\_path.sh root@hadoop02:/etc/profile.d/my\_path.sh

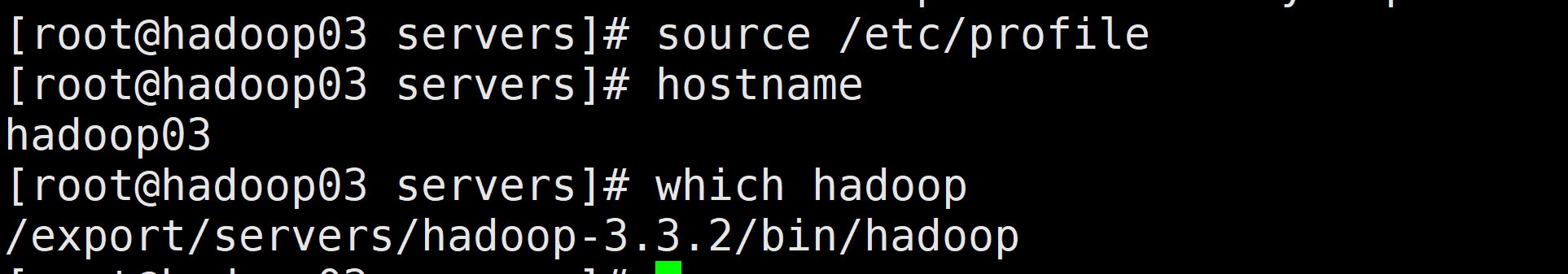
source /etc/profile

source /etc/profile



**检查配置：**





**配置成功！！！**

**分发时候出现问题：**

在装完[hadoop](https://so.csdn.net/so/search?q=hadoop&spm=1001.2101.3001.7020" \t "_blank)及jdk之后，在执行start-all.sh的时候出现

—解决方案：  
root@localhost's password:localhost:permission denied,please try again。

1.修改root密码：#sudo passwd root

2.辑配置文件，允许以 root 用户通过 [ssh](https://so.csdn.net/so/search?q=ssh&spm=1001.2101.3001.7020" \t "_blank) 登录：sudo vim /etc/ssh/sshd\_config

　找到：PermitRootLogin prohibit-password禁用

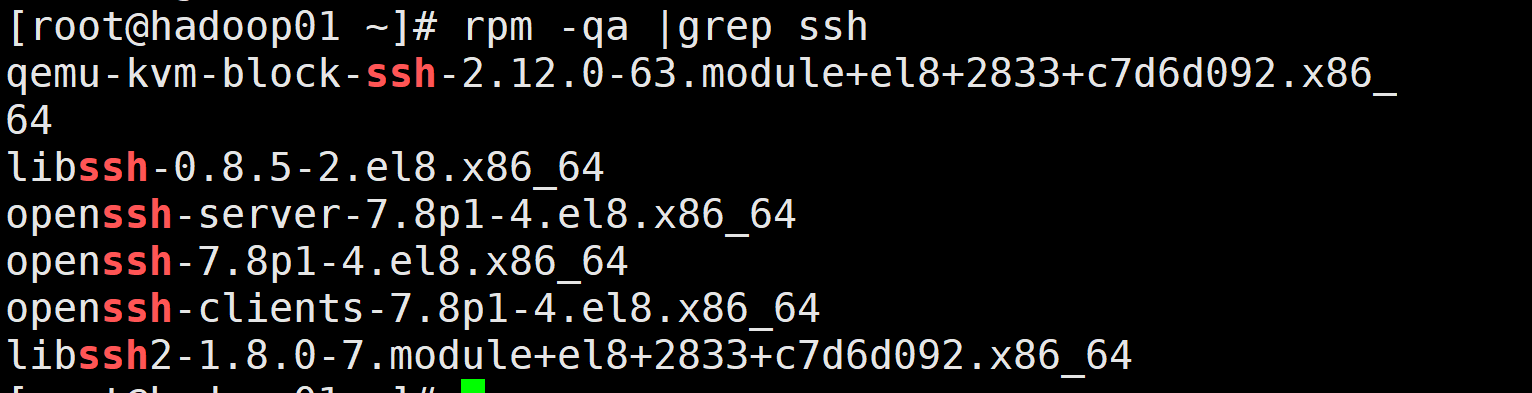
   添加：PermitRootLogin yes

3.sudo service ssh restart

# ssh免密登陆

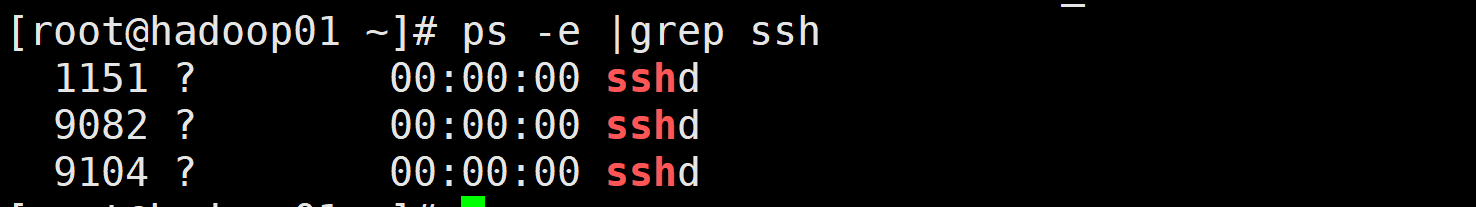
# 配置ssh

### 检查是否安装ssh



### 查看服务是否启动

**若没有安装则使用yum install openssh-server 安装**



ssh-copy-id hadoop02

**拷贝到02，实现免密登陆**

[root@mail ~]# cd /root/.ssh/  
[root@mail .ssh]# ll  
total 16  
-rw------- 1 root root 571 Mar 27 14:05 authorized\_keys  
-rw------- 1 root root 2655 Mar 27 14:06 id\_rsa  
-rw-r--r-- 1 root root 573 Mar 27 14:06 id\_rsa.pub  
-rw-r--r-- 1 root root 942 Mar 27 14:05 known\_hosts

1. authorized\_keys：授权的对象
2. id\_rsa ：私钥
3. id\_rsa.pub:公钥

### ssh免密登陆

流程分析：

第一步：在三台机器上执行一下命令生成公钥和私钥

Hadoop01: ssh-keygen -t rsa  
查看密钥对: cd .ssh/ ->ls

第二步：拷贝公钥到同一台机器，及将3个公钥拷贝到hadoop01上

Hadoop01: ssh-copy-id hadoop01 ->yes->输入01的密码

第三部：复制第一胎机器的公钥到其他机器

Hadoop01: scp /root/.ssh/authorized\_keys hadoop02:/root/.ssh ->yes->输密码  
 scp /root/.ssh/authorized\_keys hadoop03:/root/.ssh ->yes->输密码

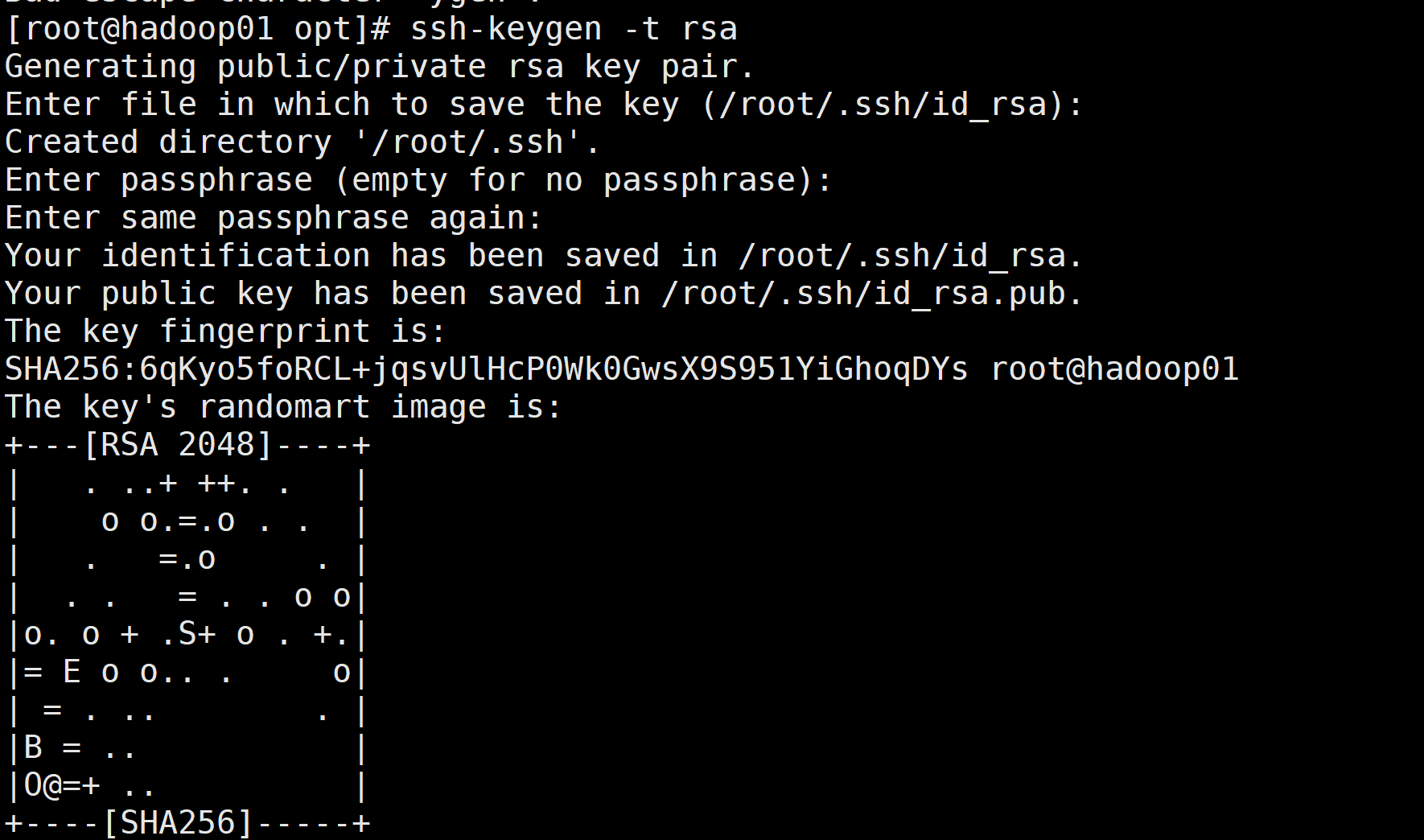
第四步：测试

Hadoop01: ssh hadoop02 链接hadoop02 exit退出

请依次在3台机器上测试

**我们先配置hadoop01**

ssh-keygen -t rsa



**此时我们可以在root下面找到ssh文件**

[root@hadoop01 .ssh]# pwd  
/root/.ssh  
[root@hadoop01 .ssh]# ls  
id\_rsa id\_rsa.pub

**我们需要将公钥拷贝到ssh**

ssh-copy-id hadoop02



ssh-copy-id hadoop03

**我们使用ssh登陆，不需要密码**

[root@hadoop01 .ssh]# ssh root@hadoop02  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last login: Mon Mar 28 01:58:36 2022 from 192.168.121.5  
[root@hadoop02 ~]#

继续分别配置免密登录，步骤重复

注意：这个是针对我们的用户的，其他用户还是需要密码

**此时我们在任何一台主机都可以操控**

## 集群配置

### 1）集群部署规划

**此时我们各个虚拟机都已经安装和配置完成**

**注意：**

**Ø NameNode和SecondaryNameNode不要安装在同一台服务器**

**Ø ResourceManager也很消耗内存，不要和NameNode、SecondaryNameNode配置在同一台机器上。**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **hadoop01** | **hadoop02** | **hadoop03** |
| HDFS | NameNode DataNode | DataNode | SecondaryNameNode DataNode |
| YARN | NodeManager | ResourceManager NodeManager | NodeManager |

### 2）配置文件说明

**Hadoop配置文件分两类：默认配置文件和自定义配置文件，只有用户想修改某一默认配置值时，才需要修改自定义配置文件，更改相应属性值。**

**（1）默认配置文件：**

|  |  |
| --- | --- |
| 要获取的默认文件 | 文件存放在Hadoop的jar包中的位置 |
| core-default.xml | hadoop-common-3.1.3.jar/core-default.xml |
| hdfs-default.xml | hadoop-hdfs-3.1.3.jar/hdfs-default.xml |
| yarn-default.xml | hadoop-yarn-common-3.1.3.jar/yarn-default.xml |
| mapred-default.xml | hadoop-mapreduce-client-core-3.1.3.jar/mapred-default.xml |

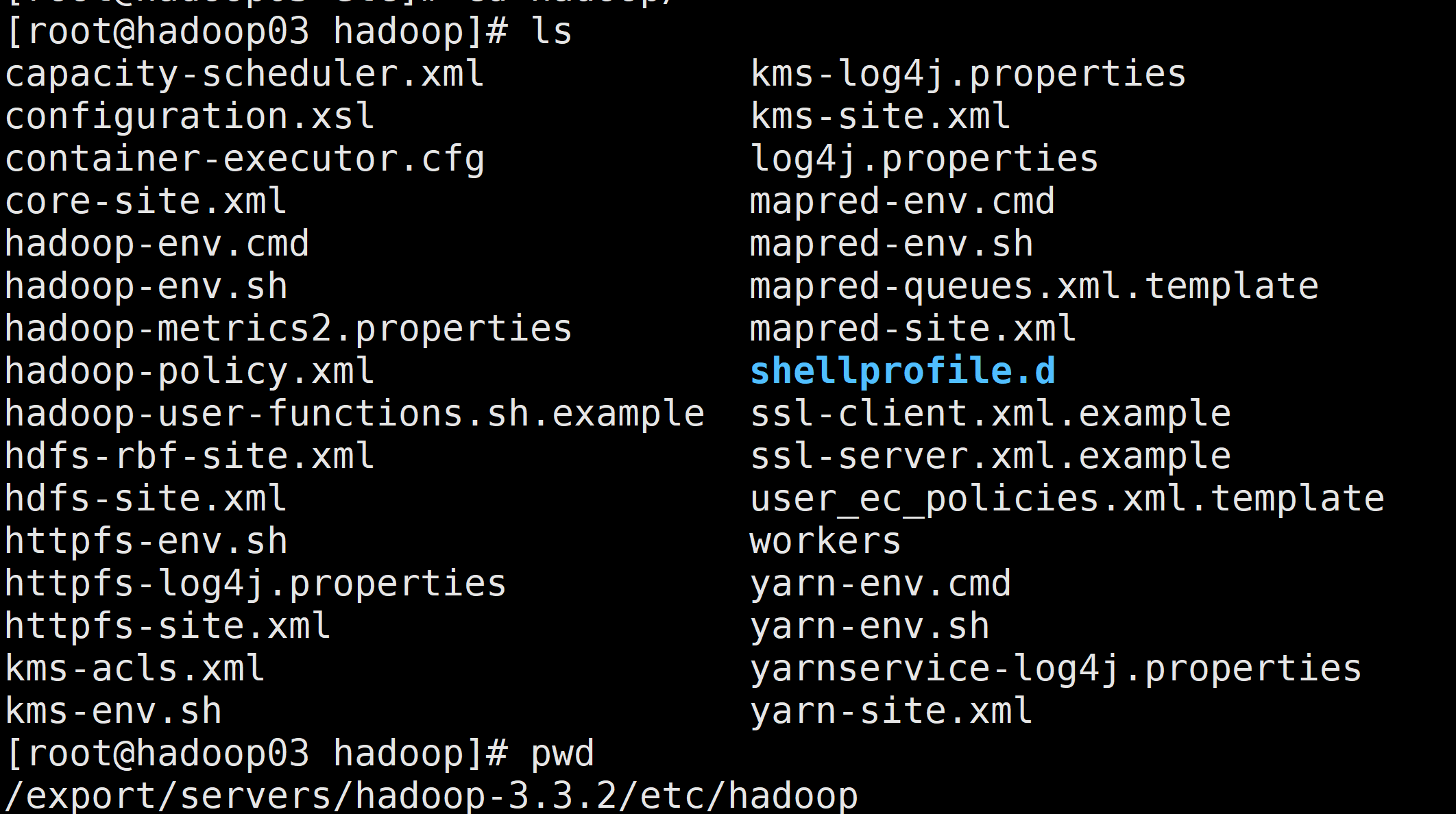
**（2）自定义配置文件：**

**core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-site.xml四个配置文件存放在$HADOOP\_HOME/etc/hadoop这个路径上，用户可以根据项目需求重新进行修改配置。**

**（$HADOOP\_HOME环境变量目录安装位置在/export/servers/hadoop-3.3.2）**

### 配置集群

**路径：cd /export/servers/hadoop-3.3.2/etc/hadoop**



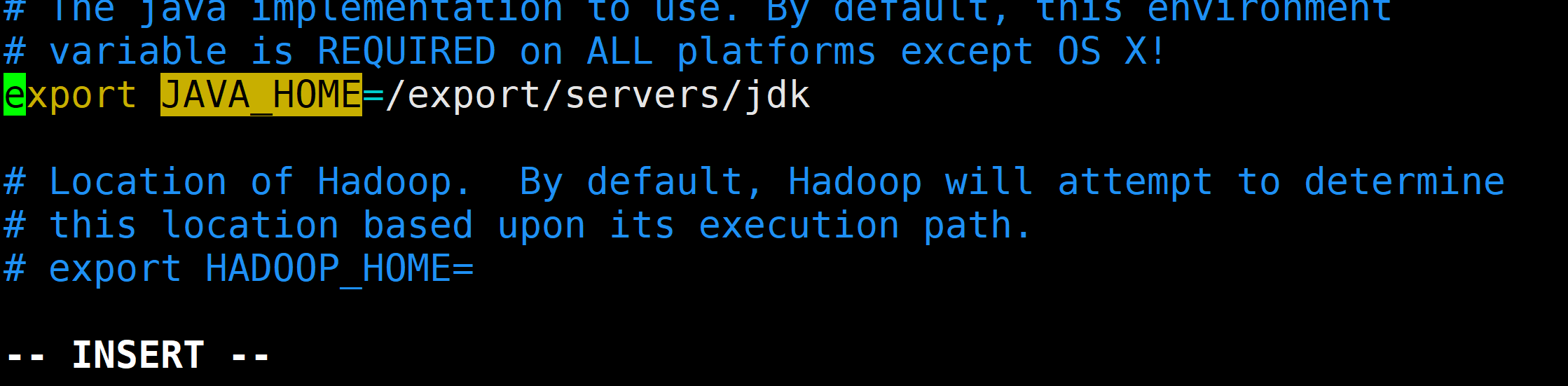
**（1）核心配置文件**

在配置文件的时候，按照上面的需求来配置

注意🖱️🖱️🖱️**xml格式对空格有要求，我们在插入的时候一定要用格式化后插入，使用vim的时候先按o在中间插入不能有空格！！！！**

1. **修改hadoop-env.sh文件**

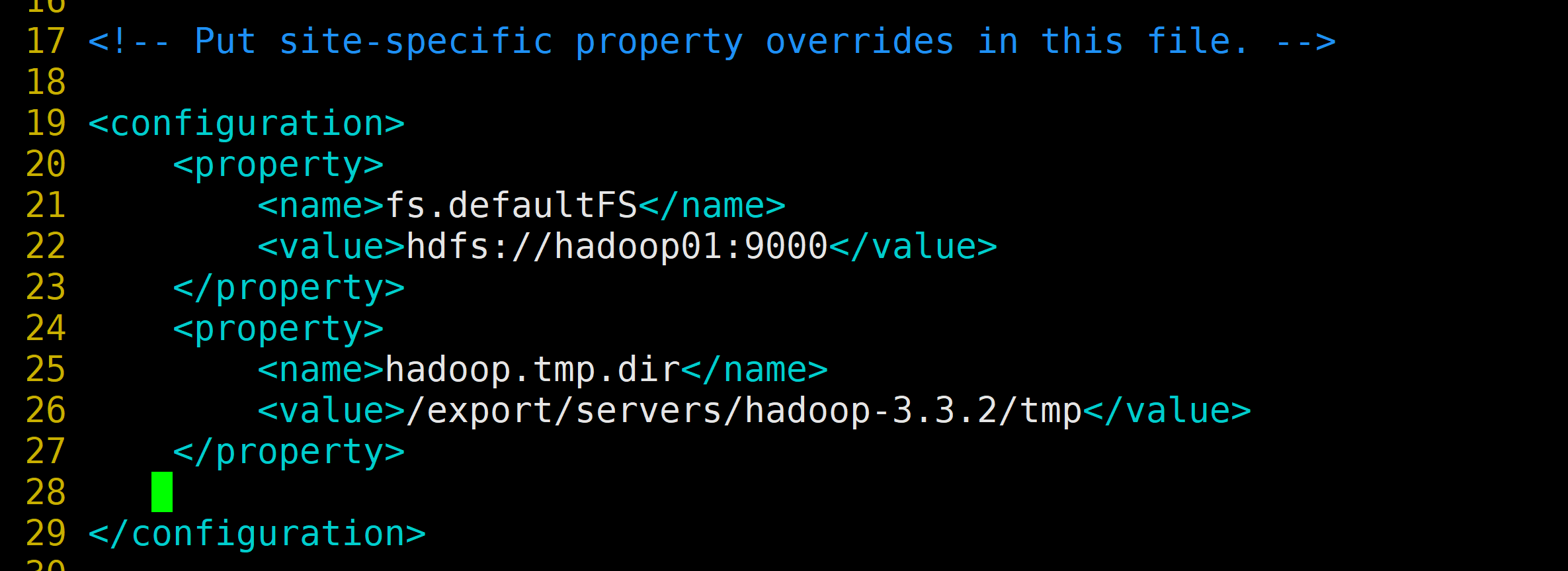
**修改：**export JAVA\_HOME=/export/servers/jdk



1. **修改core-site.xml文件**

**即使是最后一行也不要有空格或者空行**

补充：



<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoop01:9000</value>

</property>

<property>

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

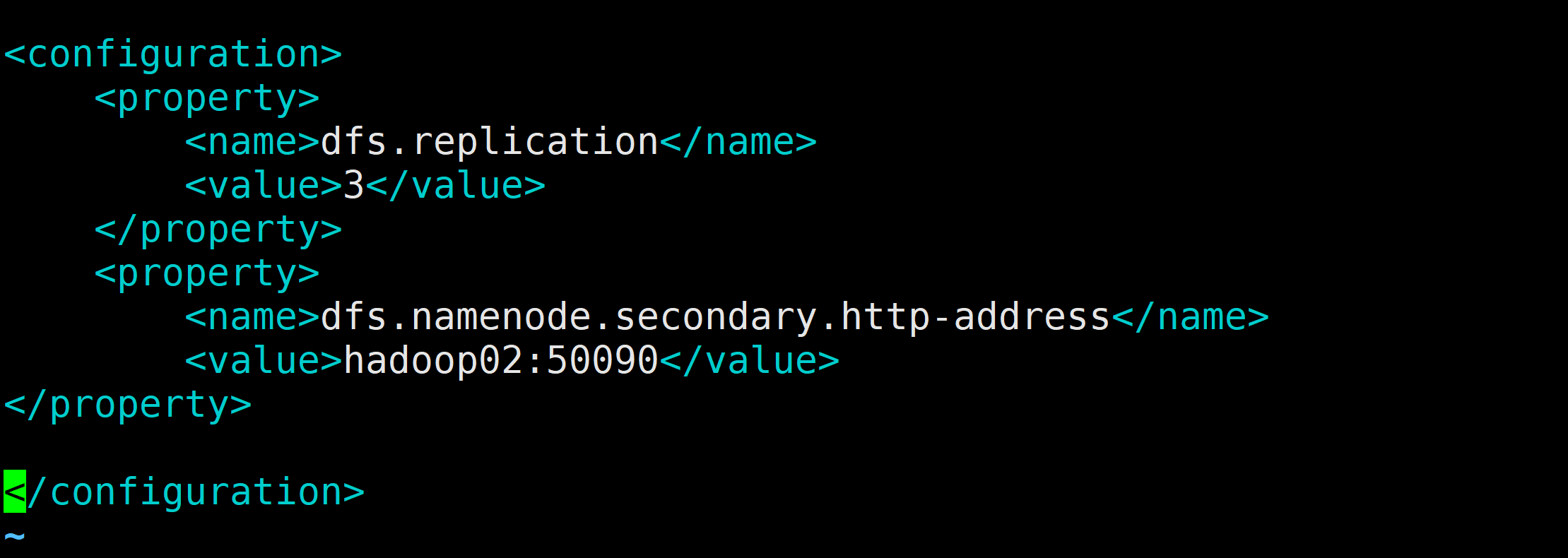
<value>/export/servers/hadoop-3.3.2/tmp</value>

</property>

</configuration>

1. **修改hdfs-site.xml文件**

补充：



<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

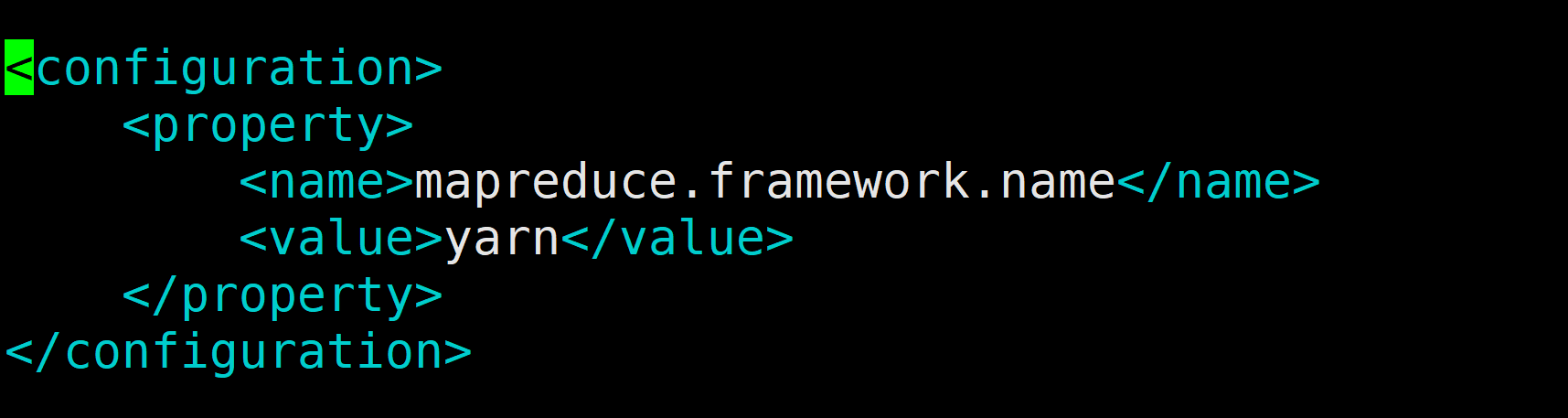
<value>hadoop02:50090</value>

</property>

</configuration>

1. **修改mapred-site.xml文件**

补充：



<configuration>

<property>

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

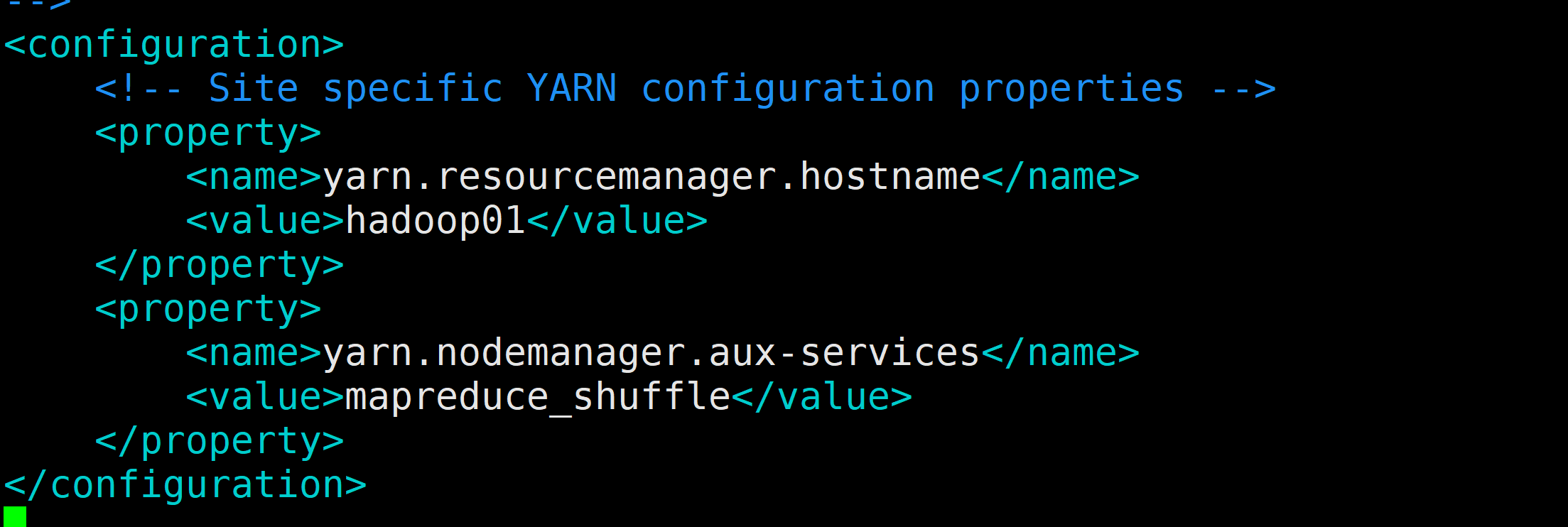
<value>yarn</value>

</property>

</configuration>

1. **修改yarn-site.xml文件**

补充：



<configuration>

<!-- Site specific YARN configuration properties -->

<property>

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

<value>hadoop01</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

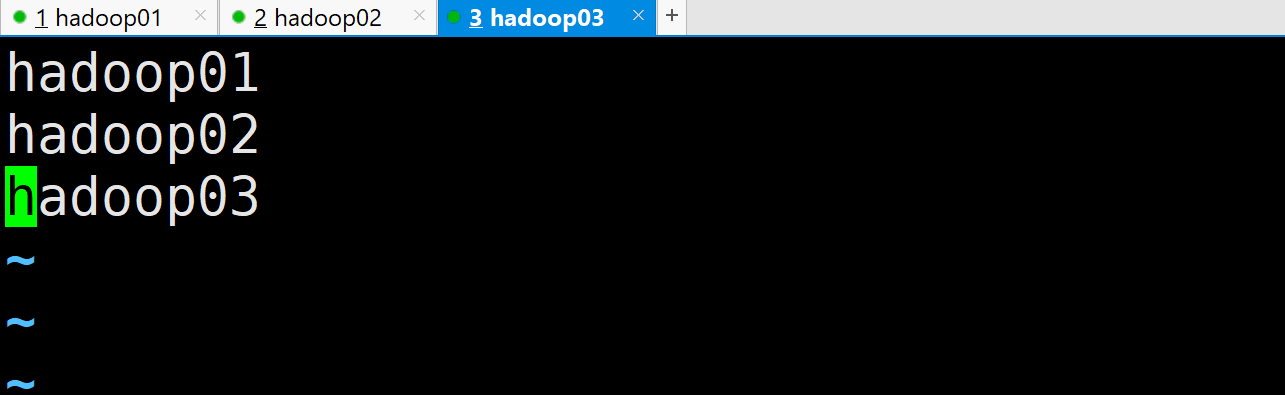
</configuration>

1. **修改workers文件,打开配置文件，删除里面内容（默认localhost）,然后配置下面内容：**

hadoop01

hadoop02

hadoop03



**这个文件里面不可以有空格，不可以有空行，否则会被认为是主机名称**

## 将集群中的节点的配置文件分发到其他节点（当前是hadoop3）

**目录：/export/servers/hadoop-3.3.2/etc/Hadoop**

**在<https://c.runoob.com/front-end/710/>网站上对其格式化后再填入**

**配置core-site.xml**

cd $HADOOP\_HOME/etc/hadoop  
  
 vim core-site.xml

文件内容如下：

<?xml version="1.0" encoding="UTF-8"?>  
  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
  
<configuration>  
  
 <!-- 指定NameNode的地址 -->  
  
 <property>  
  
 <name>fs.defaultFS</name>  
  
 <value>hdfs://hadoop01:8020</value>  
  
 </property>  
  
  
 <!-- 指定hadoop数据的存储目录 -->  
  
 <property>  
  
 <name>hadoop.tmp.dir</name>  
  
 <value>/export/servers/hadoop-3.3.2/data</value>  
  
 </property>  
  
   
  
 <!-- 配置HDFS网页登录使用的静态用户为root -->  
  
 <property>  
  
 <name>hadoop.http.staticuser.user</name>  
  
 <value>root</value>  
  
 </property>  
  
</configuration>

**（2）HDFS配置文件**

配置hdfs-site.xml

# vim hdfs-site.xml

文件内容如下：

<?xml version="1.0" encoding="UTF-8"?>  
  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
  
   
  
<configuration>  
  
 <!-- nn web端访问地址-->  
  
 <property>  
  
 <name>dfs.namenode.http-address</name>  
  
 <value>hadoop01:9870</value>  
  
 </property>  
  
 <!-- 2nn web端访问地址-->  
  
 <property>  
  
 <name>dfs.namenode.secondary.http-address</name>  
  
 <value>hadoop03:9868</value>  
  
 </property>  
  
</configuration>

（3）YARN配置文件

配置yarn-site.xml

# vim yarn-site.xml

文件内容如下：

<?xml version="1.0" encoding="UTF-8"?>  
  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
  
   
  
<configuration>  
  
 <!-- 指定MR走shuffle -->  
  
 <property>  
  
 <name>yarn.nodemanager.aux-services</name>  
  
 <value>mapreduce\_shuffle</value>  
  
 </property>  
  
   
  
 <!-- 指定ResourceManager的地址-->  
  
 <property>  
  
 <name>yarn.resourcemanager.hostname</name>  
  
 <value>hadoop02</value>  
  
 </property>  
  
   
  
 <!-- 环境变量的继承 -->  
  
 <property>  
  
 <name>yarn.nodemanager.env-whitelist</name>  
  
 <value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>  
  
 </property>  
  
</configuration>

**（4）MapReduce配置文件**

配置mapred-site.xml

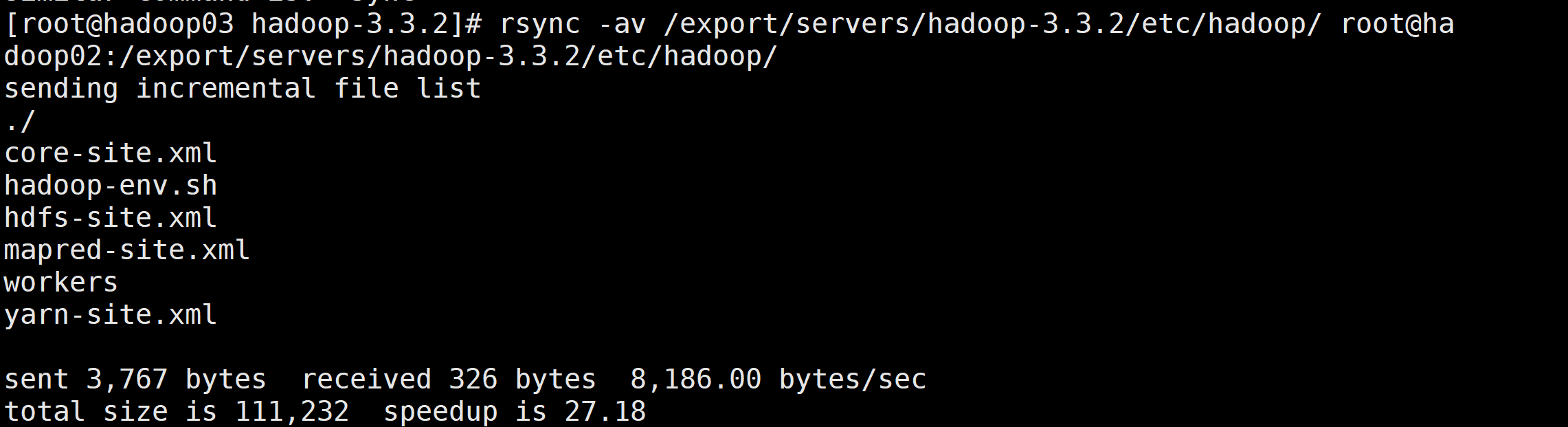
vim mapred-site.xml

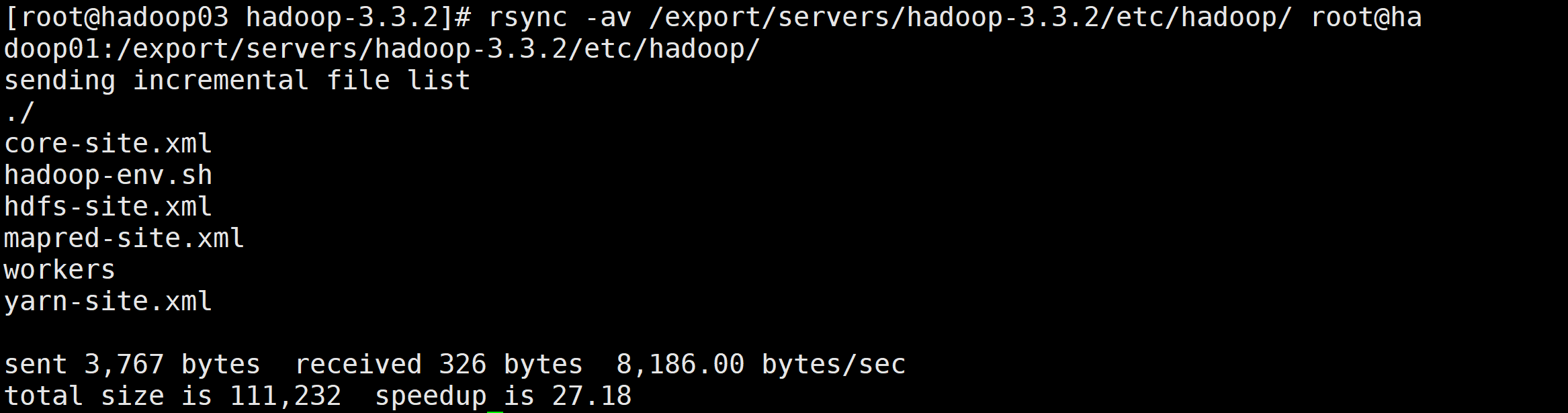
文件内容如下：

<?xml version="1.0" encoding="UTF-8"?>  
  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
  
<configuration>  
  
 <!-- 指定MapReduce程序运行在Yarn上 -->  
  
 <property>  
  
 <name>mapreduce.framework.name</name>  
  
 <value>yarn</value>  
  
 </property>  
  
</configuration>

**4）在集群上分发配置好的Hadoop配置文件**

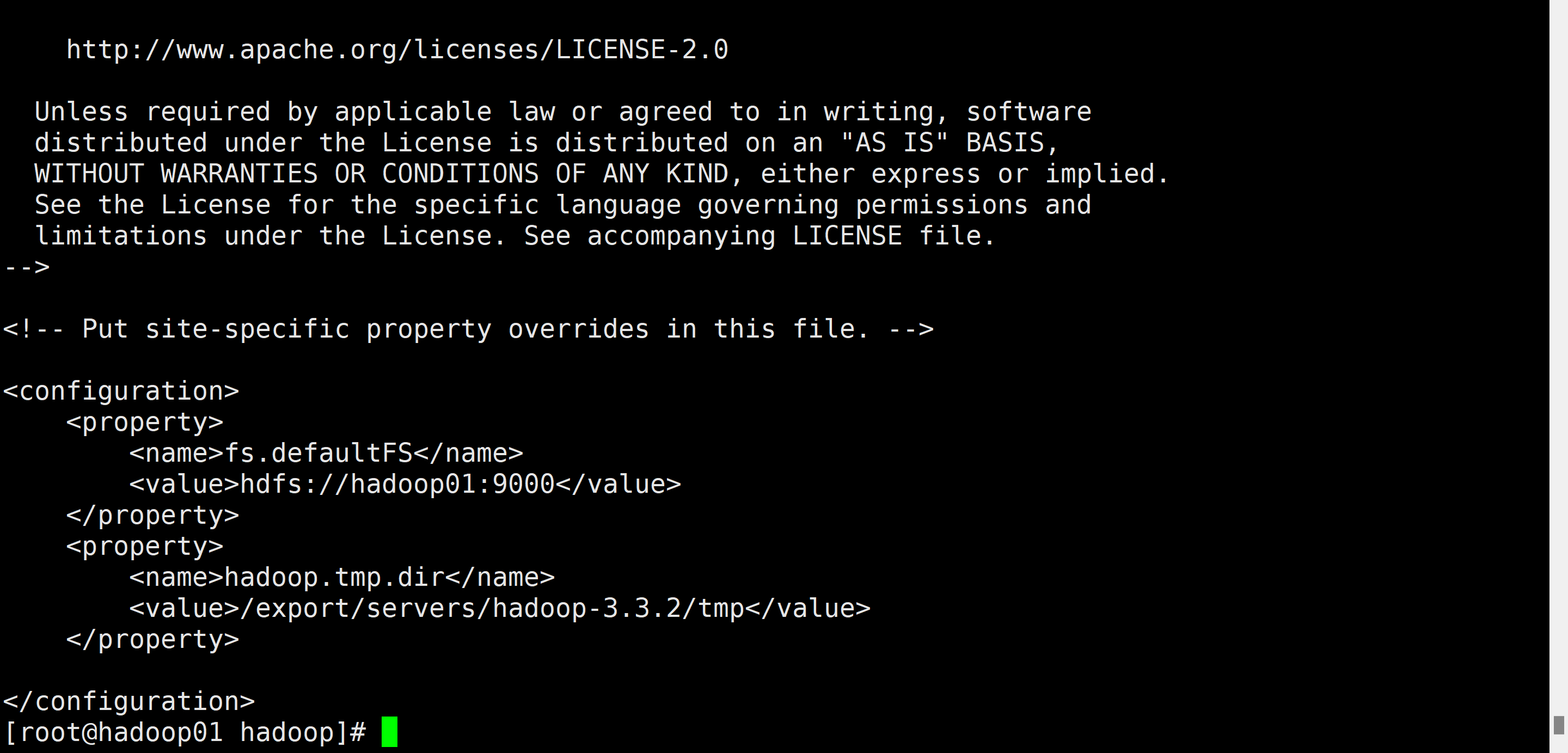
rsync -av /export/servers/hadoop-3.3.2/etc/hadoop/ root@hadoop02:/export/servers/hadoop-3.3.2/etc/hadoop/





**5）去01和02上查看文件分发情况**

cat /export/servers/hadoop-3.3.2/etc/hadoop/core-site.xml  
  
cat /export/servers/hadoop-3.3.2/etc/hadoop/core-site.xml



**成功！！！**

**分别执行source /etc/profile**

## 启动集群

1. **如果集群是第一次启动**，需要在hadoop102节点格式化NameNode（注意：格式化NameNode，会产生新的集群id，导致NameNode和DataNode的集群id不一致，集群找不到已往数据。如果集群在运行过程中报错，需要重新格式化NameNode的话，一定要先停止namenode和datanode进程，并且要删除所有机器的data和logs目录，然后再进行格式化。）

**重新格式化：**

1、删除 hadoop 三台节点配置路径下的残留文件

（1）core-site.xml

删除/usr/local/src/hadoop/tmp目录

rm -rf /export/servers/hadoop-3.3.2/data # 删除hdfs缓存文件

# hadoop.tmp.dir 配置了 hdfs数据缓存路径

<property>

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

<value>file:/usr/local/src/hadoop/tmp</value>

</property>

（2）hdfs-site.xml

删除/usr/local/src/hadoop/dfs/name目录

rm -rf /usr/local/src/hadoop/dfs/name # 删除NN缓存目录

# 配置了NN的缓存路径

<property>

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

<value>file:/usr/local/src/hadoop/dfs/name</value>

</property>

（3）hdfs-site.xml

删除/usr/local/src/hadoop/dfs/data目录

rm -rf /usr/local/src/hadoop/dfs/data # 删除DN缓存目录

# 配置了DN缓存路径

<property>

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

<value>file:/usr/local/src/hadoop/dfs/data</value>

</property>

2、手动创建配置文件

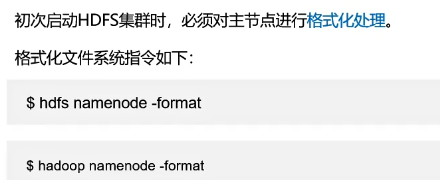
[root@master hadoop]# mkdir -p /usr/local/src/hadoop/dfs/name

[root@master hadoop]# mkdir -p /usr/local/src/hadoop/dfs/data

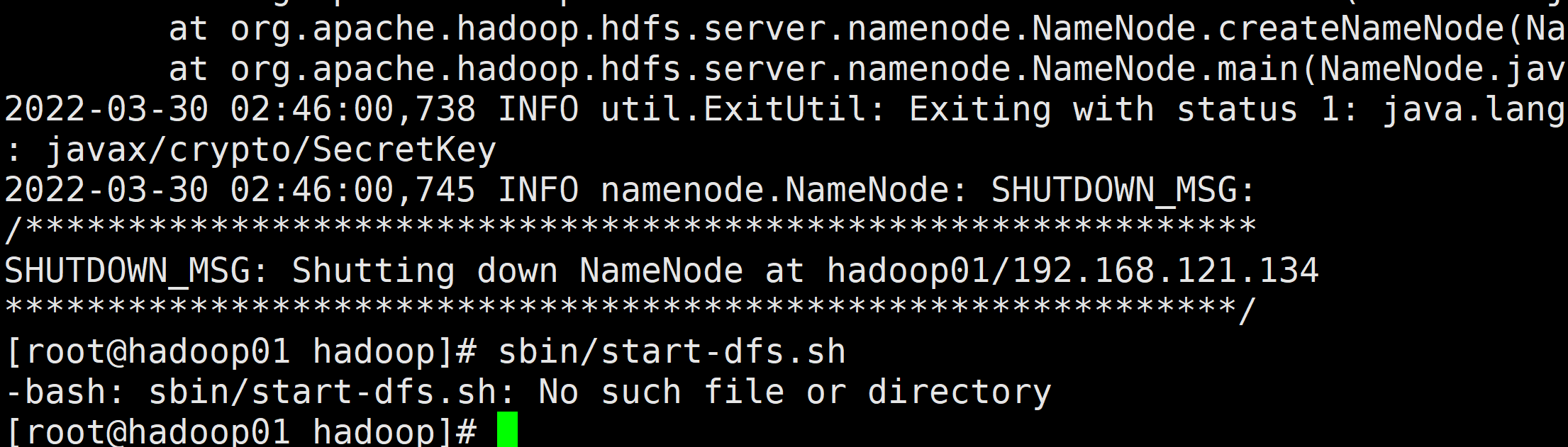
[root@master hadoop]# mkdir -p /usr/local/src/hadoop/tmp

4、启动集群后重新格式化

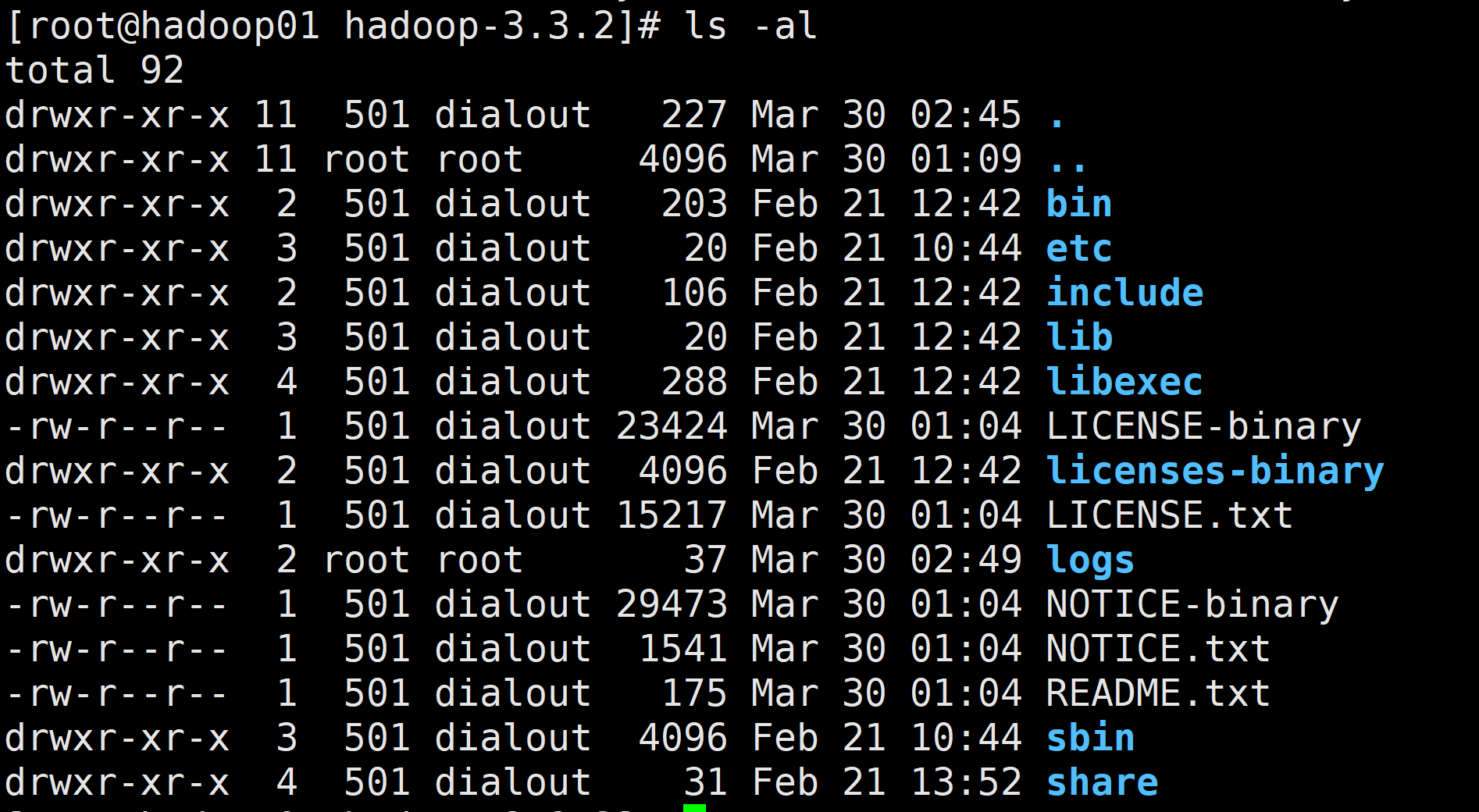
hdfs namenode -format



hdfs namenode -format



**(如果没有提醒出错那么这个时候初始化完成，这个时候多了一个路径：data/logs)**



**Hadoop namenode -format**

（2）启动HDFS

sbin/start-dfs.sh

（3）**在配置了ResourceManager的节点（hadoop02）**启动YARN

sbin/start-yarn.sh



Stutic显示为0，成功！！！！！

（4）Web端查看HDFS的NameNode

（a）浏览器中输入：<http://hadoop01:9870>

（b）查看HDFS上存储的数据信息

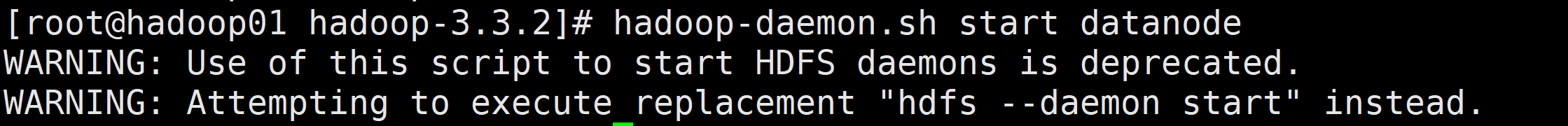
（5）Web端查看YARN的ResourceManager

（a）浏览器中输入：<http://hadoop02:8088>

（b）查看YARN上运行的Job信息



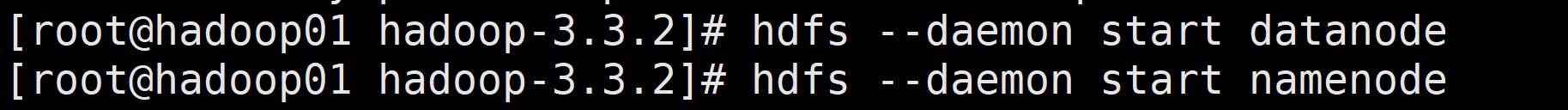
**使用hadoop-daemon.sh start namenode**



**对hadoop1：**

**[root@hadoop01 hadoop-3.3.2]# hdfs --daemon start datanode**

**[root@hadoop01 hadoop-3.3.2]# hdfs --daemon start namenode**



**对hadoop2，hadoop3：**

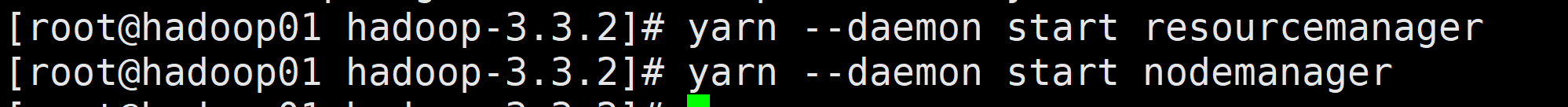
**[root@hadoop02 hadoop-3.3.2]# hdfs --daemon start datanode**

**[root@hadoop03 hadoop-3.3.2]# hdfs --daemon start datanode**

**对hadoop1：**

**yarn --daemon start resourcemanager**

**yarn --daemon start nodemanager**



### Jps:

jps 命令类似与 linux 的 ps 命令，但是它只列出系统中所有的 Java 应用程序。 通过 jps 命令可以方便地查看 Java 进程的启动类、传入参数和 Java 虚拟机参数等信息。

如果在 linux 中想查看 java 的进程，一般我们都需要 ps -ef | grep java 来获取进程 ID。  
如果只想获取 Java 程序的进程，可以直接使用 jps 命令来直接查看。

#### 参数说明

-q：只输出进程 ID  
-m：输出传入 main 方法的参数  
-l：输出完全的包名，应用主类名，jar的完全路径名  
-v：输出jvm参数  
-V：输出通过flag文件传递到JVM中的参数



**对hadoop2,hadoop03:**

**yarn --daemon start nodemanager**





**集群崩溃处理：  
1. 先杀死所有进程**

**2. [root@hadoop01 hadoop-3.3.2]# rm -rf data/ logs/**

**初始化**

**启动集群**

