# 一、Linux用户与权限

## 1. 用户和权限的基本概念

### 1.1、基本概念

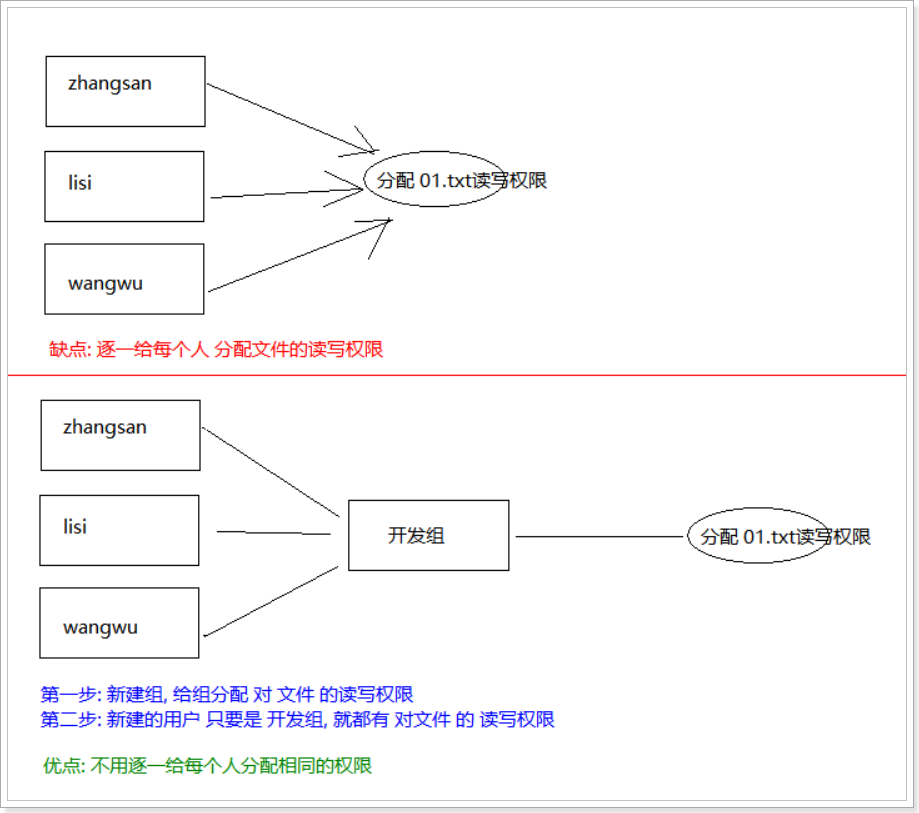
**用户** 是Linux系统工作中重要的一环, 用户管理包括 **用户** 与 **组** 管理

在Linux系统中, 不论是由本级或是远程登录系统, 每个系统都**必须拥有一个账号**, 并且**对于不同的系统资源拥有不同的使用权限**

对 **文件 / 目录** 的权限包括:

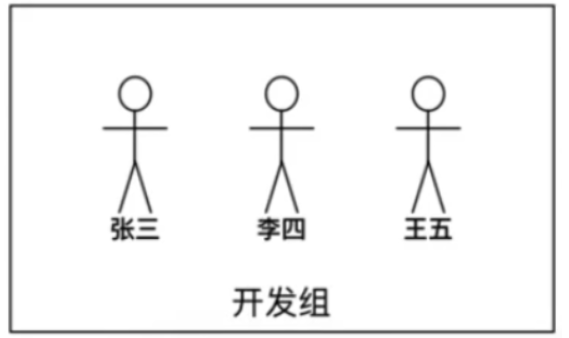
| **权限** | **英文** | **缩写** | **数字序号** |
| --- | --- | --- | --- |
| 读 | read | r | 4 |
| 写 | write | w | 2 |
| 执行 | execute | x | 1 |
| 无权限 |  | - | 0 |

在 Linux中 ,可以指定 **每一个用户** 针对 **不同的文件或者目录** 的 **不同权限**

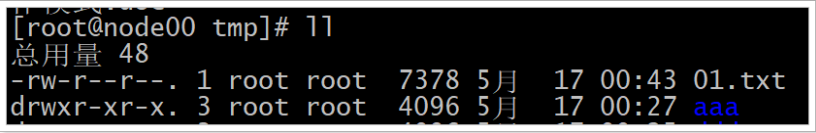


### 1.2、组

为了方便用户管理, 提出了 **组** 的概念, 如下图所示



### 1.3、ls -l扩展



ls -l 可以查看文件夹下文件的详细信息, 从左到右 依次是:

**权限**, 第一个字符如果是 d 表示目录

**硬链接数**, 通俗的讲就是有多少种方式, 可以访问当前目录和文件

**拥有者**, 家目录下 文件 / 木兰路 的拥有者通常都是 当前用户

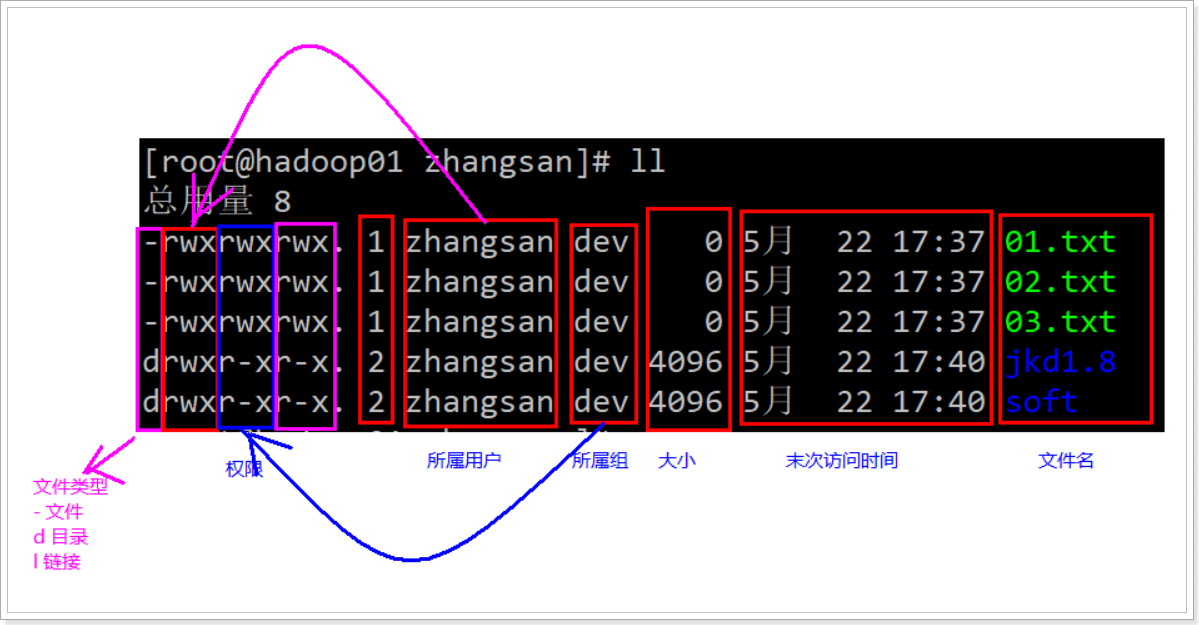
**组**, 在linux中, 很多时候, 会出现组名 和 用户名 相同的情况, 后续会讲

**大小**

**时间**

**名称**





## 2. 终端命令：组管理

**本质: 给 同一类型用户 分配权限**

| **命令** | **作用** |
| --- | --- |
| groupadd 组名 | 添加组 |
| groupdel 组名 | 删除组 |
| cat /etc/group | 确认组信息 |
| chgrp 组名 文件/目录名 | 修改文件/目录的所属组 —R递归 |

提示：

* 组信息保存在 /etc/group 文件中

我们以root:x:0:root,linuxsir 为例： 用户组root，x是密码段，表示没有设置密码，GID是0,root用户组下包括root、linuxsir以及GID为0的其它用户。

在实际应用中, 可以预先针对 **组** 设置好权限, 然后 **将不同的用户添加到对应的组中**, 从而**不用依次为每一个用户设置权限**

**演练目标：**

1、在 指定目录下 创建 aaa 目录

2、新建 dev 组

3、将 aaa 目录的组修改为 dev

chgrp dev aaa

## 3. 终端命令：用户管理

### 3.1 创建用户 、 设置密码 、删除用户

| **命令** | **作用** | **说明** |
| --- | --- | --- |
| useradd -m -g 组 新建用户名 | 添加新用户 | -m 自动建立用户家目录  -g 指定用户所在的组, 否则会建立一个和同名的组 |
| passwd 用户名 | 设置用户密码 | 如果是普通用户, 直接用 passwd可以修改自己的账号密码 |
| userdel -r 用户名 | 删除用户 | -r 选项会自动删除用户家目录 |
| cat /etc/passwd | grep 用户名 | 确认用户信息 | 新建用户后,用户信息会保存在 /etc/passwd文件夹中 |

提示:

* 创建用户时, 默认会创建一个和**用户名**同名的**组名**
* 用户信息保存在 **/etc/passwd** 文件中

/etc/passwd 文件存放的是用户的信息, 由6个分好组成的7个信息, 分别是

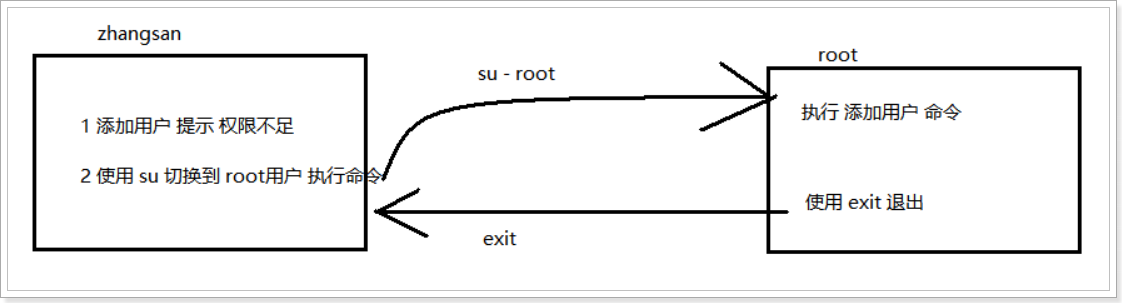


1. 用户名
2. 密码 (x, 表示加密的密码)
3. UID (用户标志)
4. GID(组标志)
5. 用户全名或本地账号
6. 家目录
7. 登录使用的Shell, 就是登录之后, 使用的终端命令

### 3.2 查看用户信息

| **命令** | **作用** |
| --- | --- |
| id [用户名] | 查看用户UID 和 GID 信息 |
| who | 查看当前所有登录的用户列表 |
| whoami | 查看当前登录用户的账户名 |

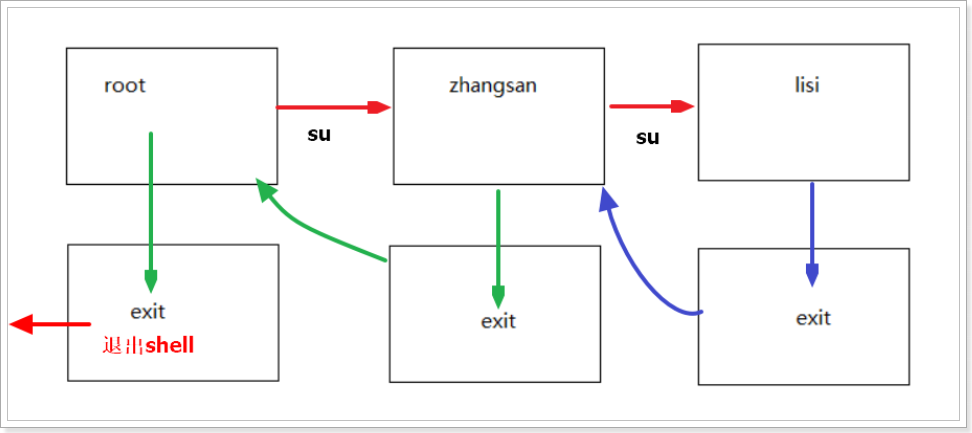
### 3.3 su切换用户



* **因为 普通用户 不能使用某些权限, 所以需要 切换用户**

| **命令** | **作用** | **说明** |
| --- | --- | --- |
| su - 用户名 | 切换用户, 并且企划目录 | -可以切换到用户家目录, 否则保持位置不变 |
| exit | 退出当前登录用户 |  |

* su 不接用户名, 可以切换到 root , 但是不推荐使用, 因为不安全
* exit 示意图如下:



### 3.4 sudo

* **虽然** 通过 su -u root 可以切换到 **root用户**, **但是** 存在严重的 **安全隐患**
* linux系统中的 root 账号通常 **用于系统的维护和管理**, 对操作系统的所有资源 **具有访问权限**
* 如果不小心使用 rm -rf ... ... , 就可能将系统搞瘫痪
* 在大多数版本的linux中, 都不推荐 **直接只用 root 账号登录系统**
* sudo 命令用来以其他身份来执行命令, 预设的身份为 root
* 用户使用 sudo 时, 必须先输入密码, 之后**5分钟的有效期限**, 超过期限则必须重新输入密码

提示: 若其未经授权的用户企图使用 sudo, 则会发出警告邮件给管理员

#### 3.4.1 给 指定用户 授予 权限

* **本质: 通知 服务器 给 特定用户 分配 临时管理员权限**

visudo

# 默认存在: root用户 具备所有的权限

root   ALL=(ALL)       ALL

​

# 授予 zhangsan 用户 所有的权限

zhangsan   ALL=(ALL)       ALL

#### 3.4.2 使用 用户 zhangsan 登录, 操作管理员命令

* 本质: **使用临时管理员权限**

# 不切换root用户, 也可以完成 添加用户的功能

sudo useradd -m -g dev zhaoliu

## 4. 修改用户权限

| **命令** | **作用** |
| --- | --- |
| chmod | 修改权限 |

准备工作: 使用root用户操作

# 清空目录中的内容

rm -rf /root/test\_tar/\*

​

# 创建测试目录

mkdir -p /root/test\_tar/aaa/

​

# 创建测试文件

touch /root/test\_tar/aaa/01.txt /root/test\_tar/aaa/02.txt

​

# 查看指定目录内容

tree /root/test\_tar

### 4.1 方式一

* chmod 可以修改 **用户/组** 对 **文件/目录** 的权限
* 命令格式如下:

chmod +/- rwx 文件名|目录名 #+表示新增权限 #-表示删除权限

提示: 已上方式会一次向修改 拥有者 / 组权限

目标演练:

# 1. 使用 root 删除目录的可读 可写 可执行权限

chmod a-rwx aaa/

​# 2. 使用 其他用户如 zhangsan 无法切换到 这个目录

# 3. 使用 root 增加目录的执行权限, 再次 使用 zhangsan 切换到目录试试

### 4.2 方式二

* 虽然 方式一 直接修改**文件|目录**的 **读|写|执行** 权限, 但是不能精确到 **拥有者|组|其他** 权限
* 命令格式如下:( u 表示所属用户 / g 表示所属组 / o 表示其他)

chmod -R u=rwx,g=rx,o=rwx 文件|目录

| **序号** | **权限** | **英文** | **缩写** | **数字序号** |
| --- | --- | --- | --- | --- |
| 01 | 读 | read | r | 4 |
| 02 | 写 | write | w | 2 |
| 03 | 执行 | execute | x | 1 |
| 04 | 无权限 |  | - | 0 |

### 4.3 方式三

* **第一个数字** 是 **拥有者权限**, **第二个数字** 是 **组权限**, **第三个数字** 是 **其他用户权限**



# 二、Linux系统信息

本节内容主要是为了方便通过远程终端维护服务器时, 查看服务器上当前 **系统日期和时间 / 磁盘空间占用情况 /程序执行情况**

本小结学习终端命令都是查询命令, 通过这些命令对系统资源的使用情况有个了解

## 1. 时间和日期

### 1.1. date时间

| **命令** | **作用** |
| --- | --- |
| date | 查看系统时间(**默认**) |
| date +"%Y-%m-%d %H:%M:%S" | 查看系统时间(**指定格式**) |
| date -s "时间字符串" | 设置系统时间 |

**第一步: 显示当前时间**

# 显示时间

date

# 按照指定格式显示时间

date +"%Y-%m-%d %H:%M:%S"

**第二步：设置系统时间**

date -s "时间字符串"

### 1.2. cal日历

| **命令** | **作用** |
| --- | --- |
| cal | 查看当前月的日历 |
| cal -y | 查看当前年的日历 |
| cal 2020 | 查看2020年的日历 |
| cal 10 2020 | 查看2020年10月的日历 |

## 2. 磁盘、内存信息

| **命令** | **作用** |
| --- | --- |
| df -h | disk free 显示磁盘剩余空间 |
| du -h [目录名] | disk usage 显示目录下的目录大小 |
| free -h | 显示内存使用情况 |

选项说明

| **参数** | **含义** |
| --- | --- |
| -h | 以人性化的方式显示文件的大小（把字节换算为相应的kb/mb/gb） |

## 3. 进程信息

| **命令** | **作用** |
| --- | --- |
| ps aux | process status 查看进程的详细情况 |
| top | 动态显示运行中进程并且排序 |
| kill [-9] 进程代号 | 终止指定代号的进程 -9 表示强行终止 |

ps 默认只会显示当前用户通过终端启动的应用程序

Linux下显示系统进程的命令ps，最常用的有ps -ef 和ps aux。这两个到底有什么区别呢？两者没太大差别，讨论这个问题，要追溯到Unix系统中的两种风格，System Ｖ风格和BSD 风格，ps aux最初用到Unix Style中，而ps -ef被用在System V Style中，两者输出略有不同。现在的大部分Linux系统都是可以同时使用这两种方式的。

ps 选项说明功能

| **选项** | **含义** |
| --- | --- |
| a | 显示终端上的所有进程,包括其他用户的进程 |
| u | 显示进程的详细状态 |
| x | 显示没有控制终端的进程 |

提示: 使用kill命令时, 最好只终止由当前用户开启的进程, 而不要终止root身份开启的进程, 否则可能导致系统崩溃

要退出 top 可以直接输入 q

# 三、Linux软件安装

## 1. rpm软件包管理器

rpm（英文全拼：redhat package manager） 原本是 Red Hat Linux 发行版专门用来管理 Linux 各项软件包的程序，由于它遵循GPL规则且功能强大方便，因而广受欢迎。逐渐受到其他发行版的采用。RPM 套件管理方式的出现，让Linux易于安装升级，间接提升了 Linux的适用度。

命令格式： rpm -参数 软件包名

| **参数** | **英文** | **含义** |
| --- | --- | --- |
| -q | query | 使用询问模式，当遇到任何问题时，rpm指令会先询问用户。 |
| -a | all | 查询所有软件包 |
| -i | info | 显示软件包的概要信息 |
| -l | list | 显示软件包中的文件列表 |
| -f | file | 文件，显示文件对应rpm包(查找文件所属的rpm包) |

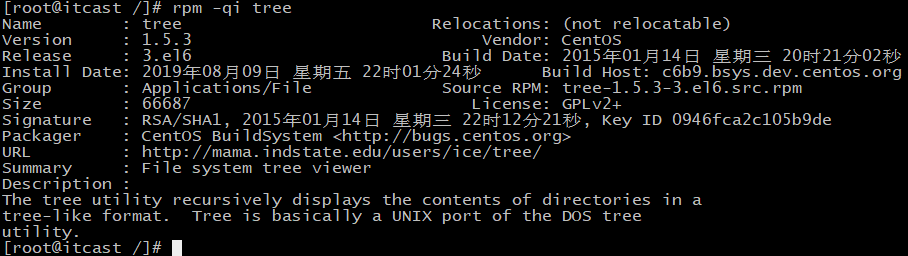
示例1：查询已安装的rpm列表，通常通过管道命令配合grep进行查找

rpm -qa | grep tree



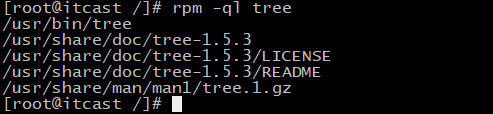
示例2：查询软件包信息

rpm -qi 软件包名称



示例3：显示软件包中的文件列表

rpm -ql 软件包名称



示例4：查看指定文件归属于哪个软件包

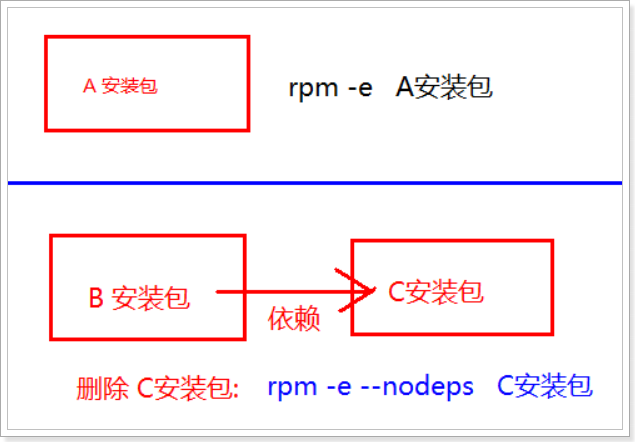
rpm -qf 文件绝对路径



### 1.1. rpm包的卸载

rpm卸载命令：

| **命令** | **英文** | **含义** |
| --- | --- | --- |
| rpm -e 软件包名称 | erase清除 | 卸载rpm软件包 |
| rpm -e --nodeps 软件包名称 | don’t check dependencies | 卸载前跳过依赖检查 |



示例：卸载tree

[root@itheima ~]# rpm -qa | grep tree

tree-1.5.3-3.el6.x86\_64

[root@itheima ~]# rpm -e --nodeps tree-1.5.3-3.el6.x86\_64

[root@itheima ~]#

[root@itheima ~]# tree /home

-bash: /usr/bin/tree: No such file or directory

[root@itheima ~]#

### 1.2. rpm包的安装

rpm包安装命令：

| **命令** | **含义** |
| --- | --- |
| rpm -ivh rpm包名称 | 安装rpm包 |

安装命令参数：

| **参数** | **英文** | **含义** |
| --- | --- | --- |
| -i | install | 安装 |
| -v | verbose | 打印提示信息 |
| -h | hase | 显示安装进度 |

## 2. yum软件包管理器

yum（ Yellow dog Updater, Modified）是一个在 Fedora 和 RedHat中的 Shell 前端软件包管理器。基于RPM包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令，而且命令简洁而又好记。

yum 语法：

yum [options] [command] [package ...]

options：可选，选项包括-h（帮助），-y（当安装过程提示选择全部为 "yes"），-q（不显示安装的过程）等等、command：要进行的操作、package：安装的包名。

### 2.1. yum常用命令

1. 列出所有可更新的软件清单命令：yum check-update

2. 更新所有软件命令：yum update

3. 仅安装指定的软件命令：yum install <package\_name>

4. 仅更新指定的软件命令：yum update <package\_name>

5. 列出所有可安裝的软件清单命令：yum list

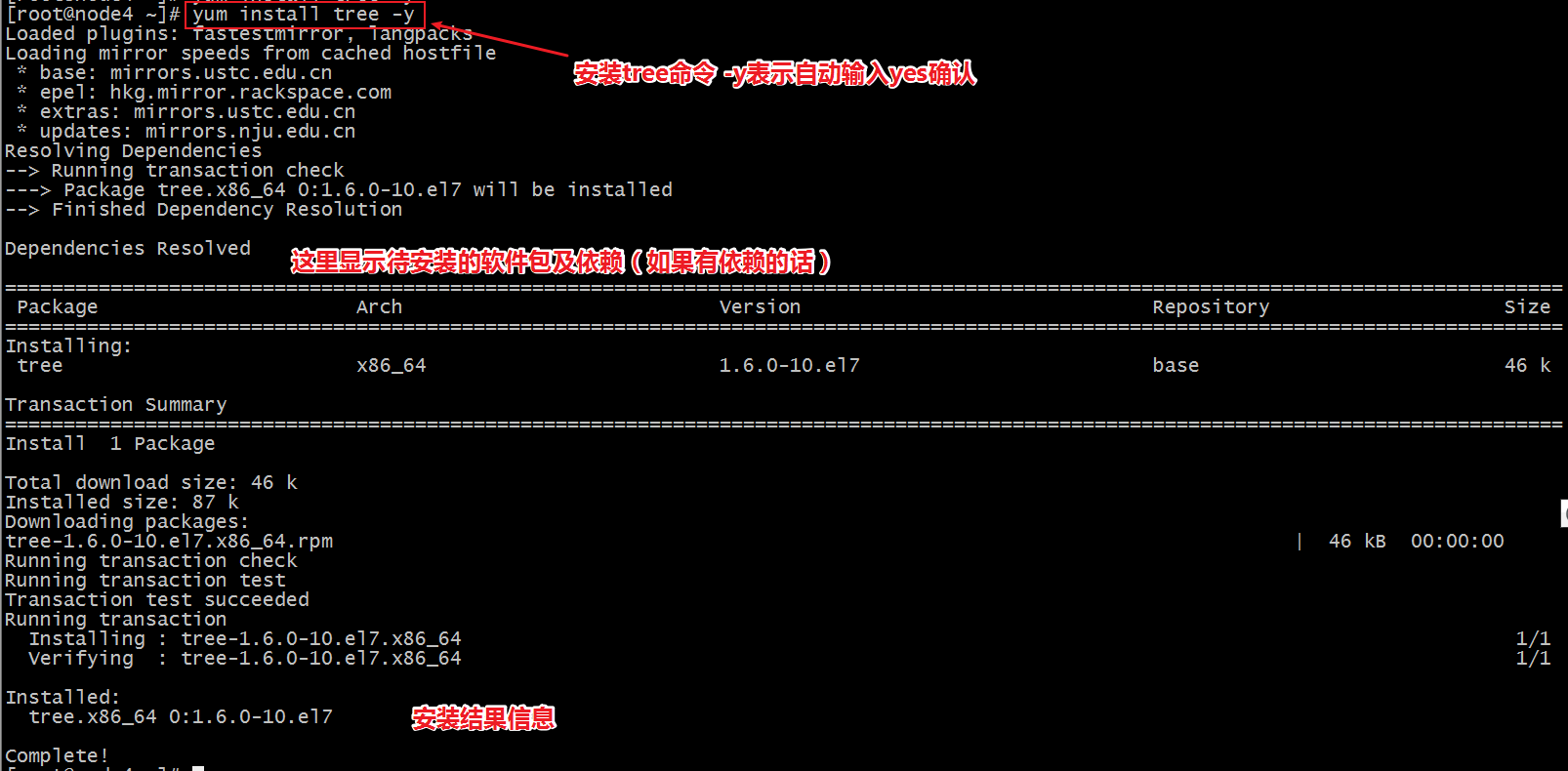
6. 删除软件包命令：yum remove <package\_name>

7. 查找软件包命令：yum search <keyword>

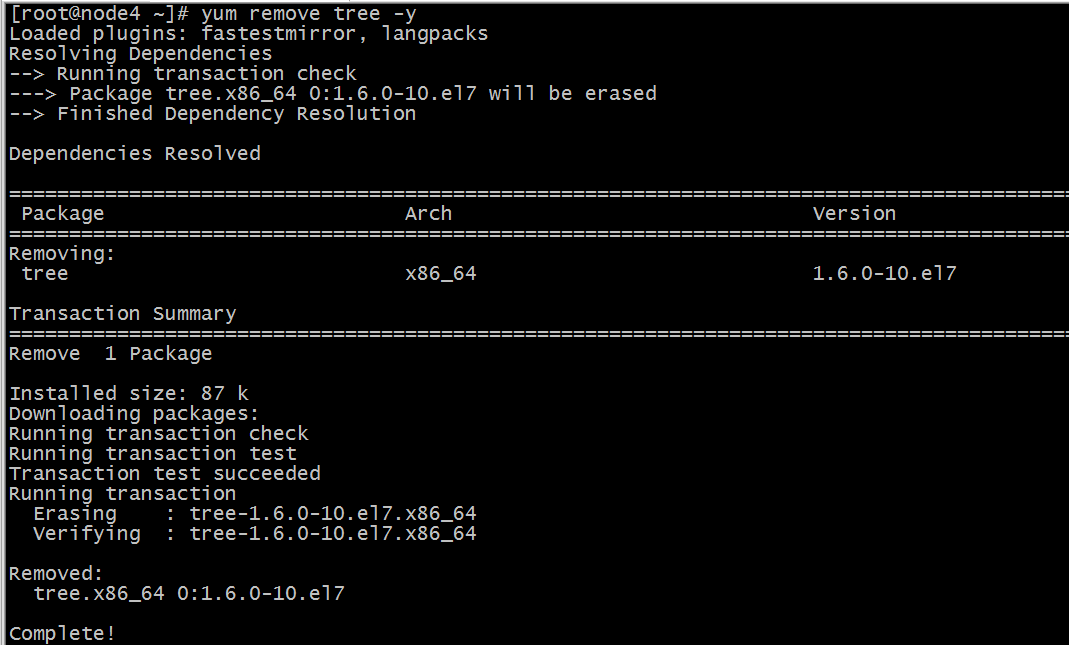
8. 列出当前可用仓库信息：yum repolist all

9. 清楚仓库缓存信息：yum clean all

### 2.2. yum在线安装软件案例



### 2.3. yum在线删除软件案例



## 3. Linux安装JDK

### 3.1. 统一软件安装路径

理论上来说，可以把软件安装在Linux的任意路径下，但是企业为了规范化管理，通常会设置约束相关的目录用来安装软件、存储软件运行的数据、日志。

|  |
| --- |
| /export/server #软件安装目录  /export/software #安装包的目录  /export/data #软件运行数据保存的目录  /export/logs  mkdir -p /export/server  mkdir -p /export/software  mkdir -p /export/data |

### 3.2. JDK安装

|  |
| --- |
| #上传安装包到/export/server下  jdk-8u241-linux-x64.tar.gz  #解压到当前目录  tar zxvf jdk-8u241-linux-x64.tar.gz  #删除红色安装包（可选）  rm -rf jdk-8u241-linux-x64.tar.gz  #配置环境变量  vim /etc/profile #G + o  export JAVA\_HOME=/export/server/jdk1.8.0\_241  export PATH=$PATH:$JAVA\_HOME/bin  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  #重新加载环境变量文件 让配置生效  source /etc/profile  [root@node1 ~]# java -version  java version "1.8.0\_241"  Java(TM) SE Runtime Environment (build 1.8.0\_241-b07)  Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode) |

## 4. Linux安装、卸载MySQL

### 4.1. 安装MySQL

step1:安装MySQL:

|  |
| --- |
| mkdir /export/software/mysql  #上传mysql-5.7.29-1.el7.x86\_64.rpm-bundle.tar 到上述文件夹下 解压  tar xvf mysql-5.7.29-1.el7.x86\_64.rpm-bundle.tar  #执行安装  yum -y install libaio  [root@node3 mysql]# rpm -ivh mysql-community-common-5.7.29-1.el7.x86\_64.rpm mysql-community-libs-5.7.29-1.el7.x86\_64.rpm mysql-community-client-5.7.29-1.el7.x86\_64.rpm mysql-community-server-5.7.29-1.el7.x86\_64.rpm  warning: mysql-community-common-5.7.29-1.el7.x86\_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY  Preparing... ################################# [100%]  Updating / installing...  1:mysql-community-common-5.7.29-1.e################################# [ 25%]  2:mysql-community-libs-5.7.29-1.el7################################# [ 50%]  3:mysql-community-client-5.7.29-1.e################################# [ 75%]  4:mysql-community-server-5.7.29-1.e################ ( 49%) |

step2:初始化MySQL:

|  |
| --- |
| #初始化  mysqld --initialize  #更改所属组  chown mysql:mysql /var/lib/mysql -R  #启动mysql  systemctl start mysqld.service  #查看生成的临时root密码  cat /var/log/mysqld.log  [Note] A temporary password is generated for root@localhost: o+TU+KDOm004 |

step3:修改root密码 授权远程访问 设置开机自启动

|  |
| --- |
| [root@node2 ~]# mysql -u root -p  Enter password: #这里输入在日志中生成的临时密码  Welcome to the MySQL monitor. Commands end with ; or \g.  Your MySQL connection id is 3  Server version: 5.7.29  Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its  affiliates. Other names may be trademarks of their respective  owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql>  #更新root密码 设置为hadoop  mysql> alter user user() identified by "hadoop";  Query OK, 0 rows affected (0.00 sec)  #授权  mysql> use mysql;  mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'hadoop' WITH GRANT OPTION;  mysql> FLUSH PRIVILEGES;  #mysql的启动和关闭 状态查看  systemctl stop mysqld  systemctl status mysqld  systemctl start mysqld  #建议设置为开机自启动服务  [root@node2 ~]# systemctl enable mysqld  Created symlink from /etc/systemd/system/multi-user.target.wants/mysqld.service to /usr/lib/systemd/system/mysqld.service.  #查看是否已经设置自启动成功  [root@node2 ~]# systemctl list-unit-files | grep mysqld  mysqld.service enabled |

### 4.2. 卸载MySQL

|  |
| --- |
| #关闭mysql服务  systemctl stop mysqld.service  #查找安装mysql的rpm包  [root@node3 ~]# rpm -qa | grep -i mysql  mysql-community-libs-5.7.29-1.el7.x86\_64  mysql-community-common-5.7.29-1.el7.x86\_64  mysql-community-client-5.7.29-1.el7.x86\_64  mysql-community-server-5.7.29-1.el7.x86\_64  #卸载  [root@node3 ~]# yum remove mysql-community-libs-5.7.29-1.el7.x86\_64 mysql-community-common-5.7.29-1.el7.x86\_64 mysql-community-client-5.7.29-1.el7.x86\_64 mysql-community-server-5.7.29-1.el7.x86\_64  #查看是否卸载干净  rpm -qa | grep -i mysql  #查找mysql相关目录 删除  [root@node1 ~]# find / -name mysql  /var/lib/mysql  /var/lib/mysql/mysql  /usr/share/mysql  [root@node1 ~]# rm -rf /var/lib/mysql  [root@node1 ~]# rm -rf /var/lib/mysql/mysql  [root@node1 ~]# rm -rf /usr/share/mysql  #删除默认配置 日志  rm -rf /etc/my.cnf  rm -rf /var/log/mysqld.log |

# 四、大数据集群服务器搭建

## 1. 新增Linux服务器

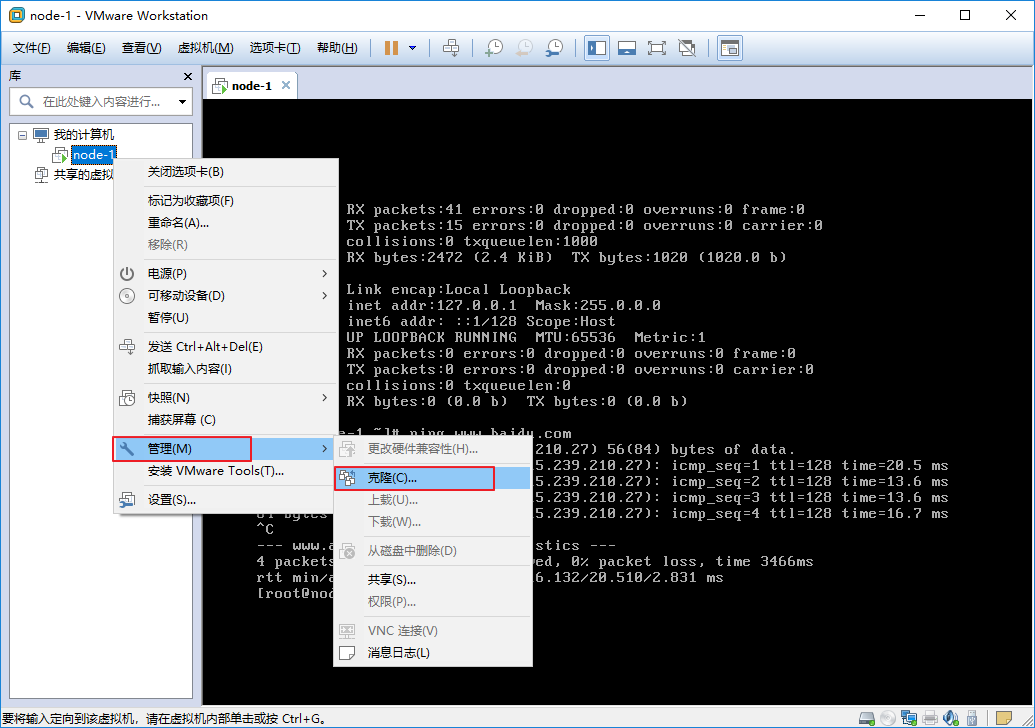
### 1.1、克隆虚拟机

学习环境中，一般使用VMware虚拟机克隆Linux系统，用来进行集群服务器的搭建。

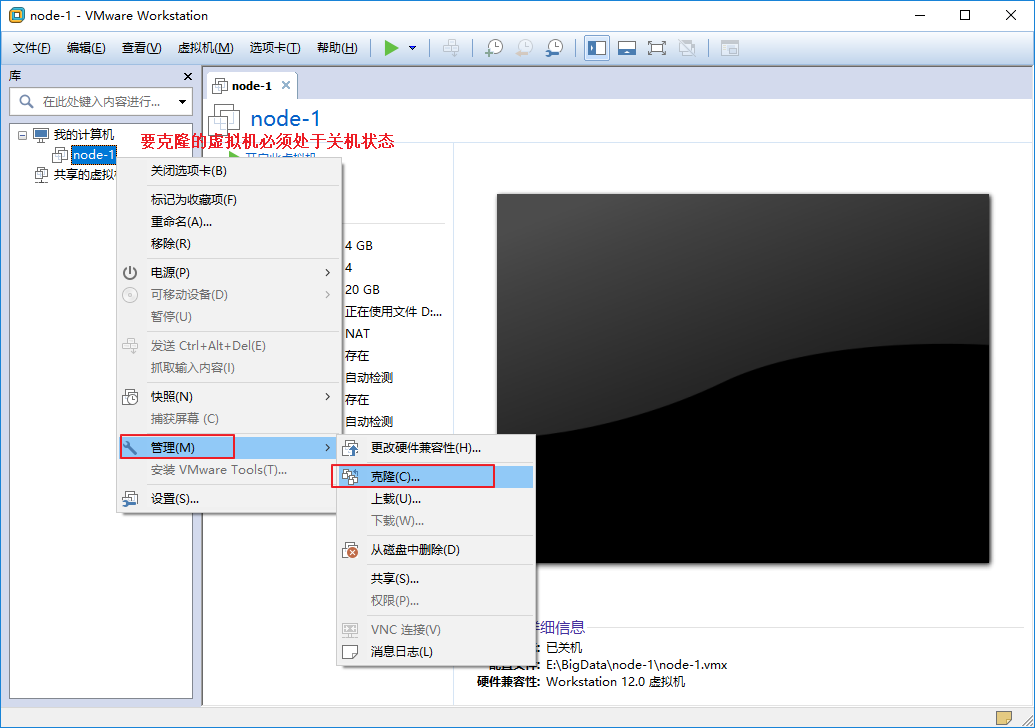
VMware支持两种类型的克隆： 完整克隆、链接克隆

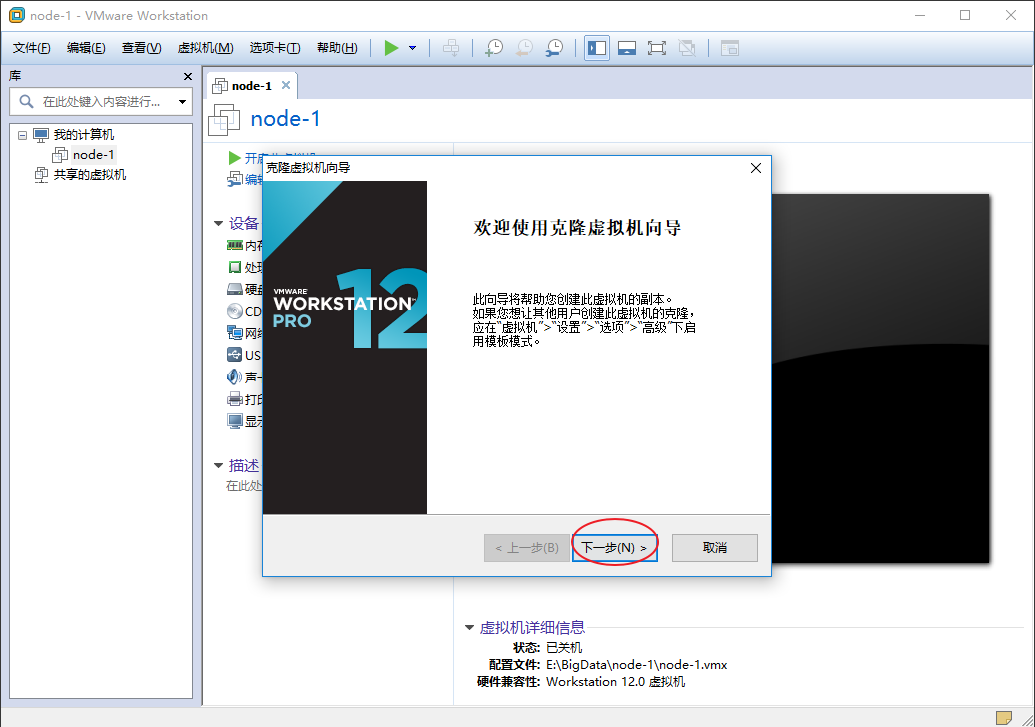
完整克隆是和原始虚拟机完全独立的一个复制，它不和原始虚拟机共享任何资源。可以脱离原始虚拟机独立使用。 链接克隆需要和原始虚拟机共享同一虚拟磁盘文件，不能脱离原始虚拟机独立运行。但采用共享磁盘文件却大大缩短了创建克隆虚拟机的时间，同时还节省了宝贵的物理磁盘空间。通过链接克隆，可以轻松的为不同的任务创建一个独立的虚拟机。

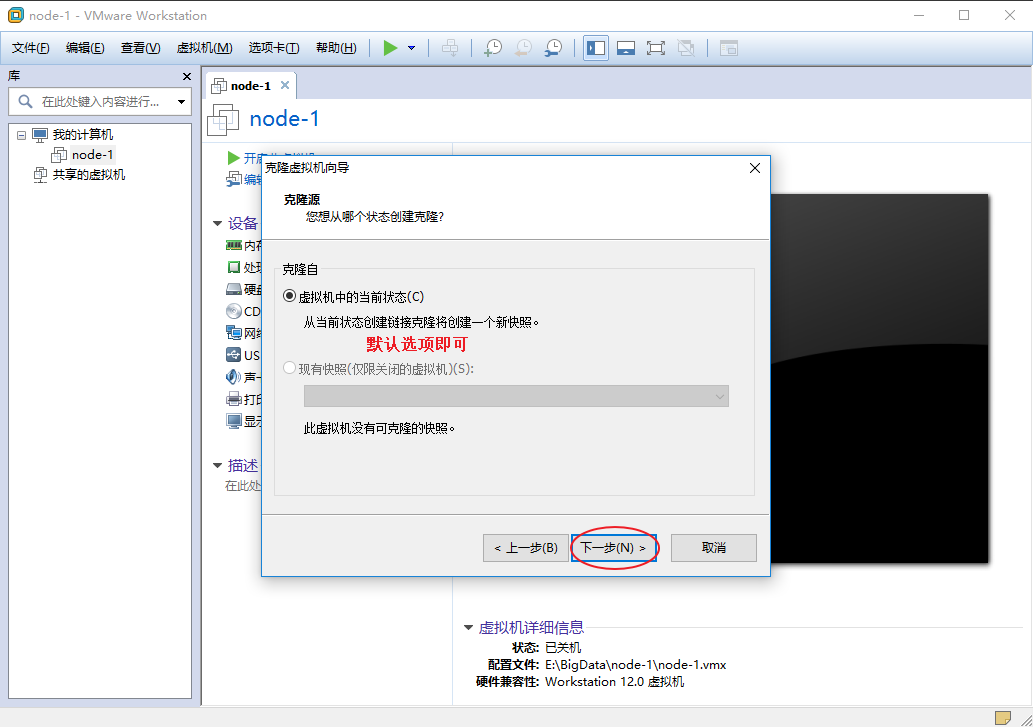
虚拟机克隆的前置条件：虚拟机处于关闭状态。

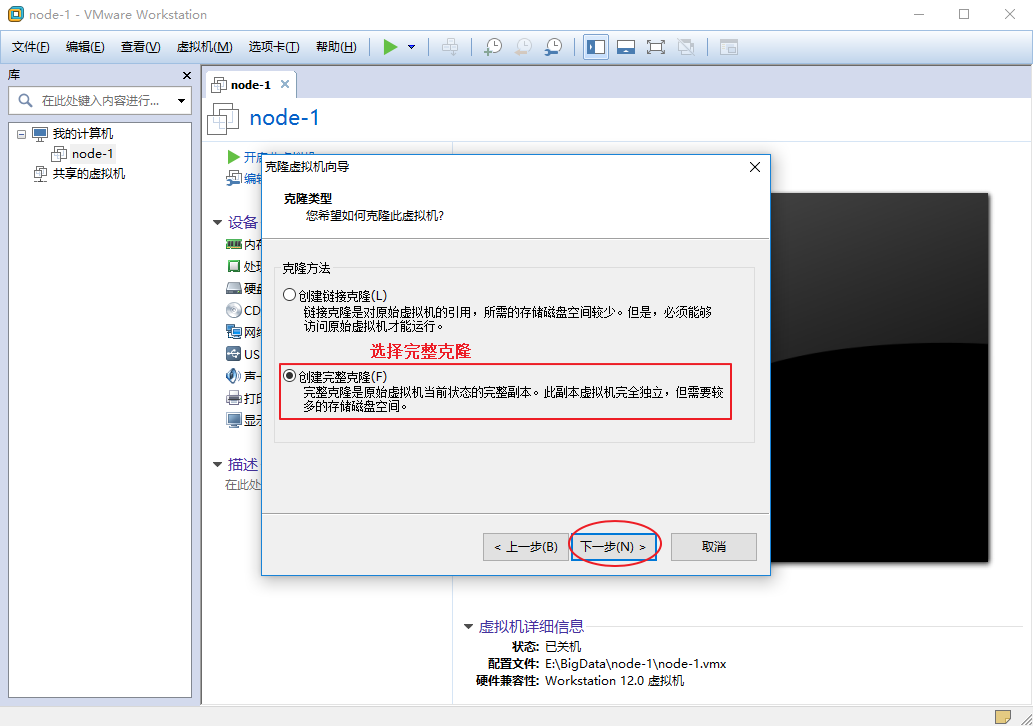


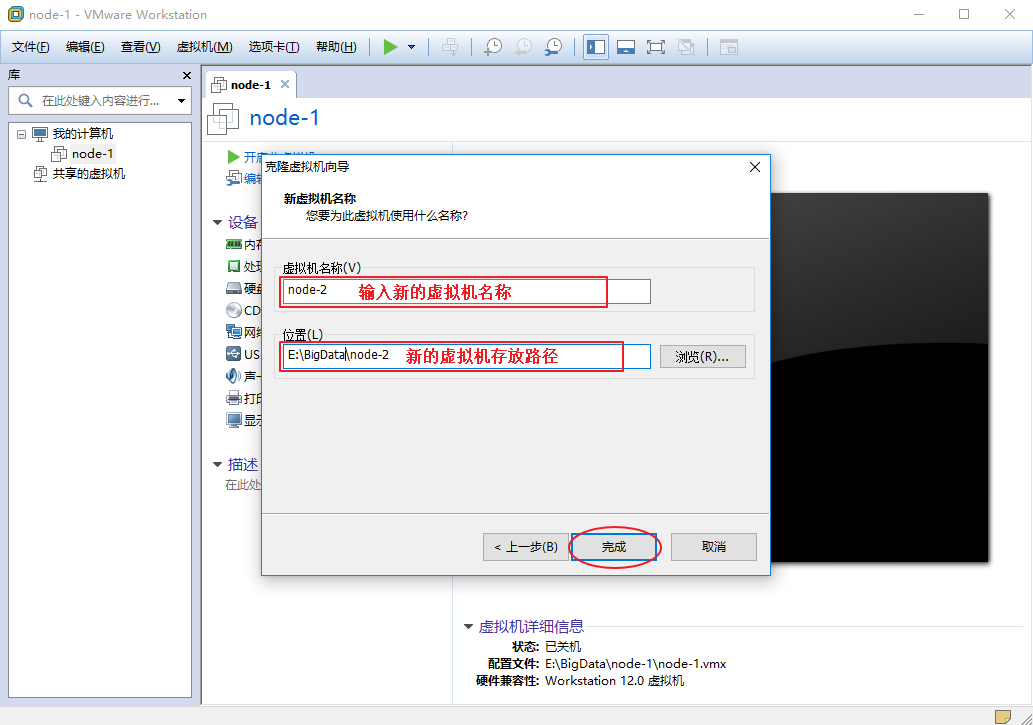
虚拟机克隆的流程：

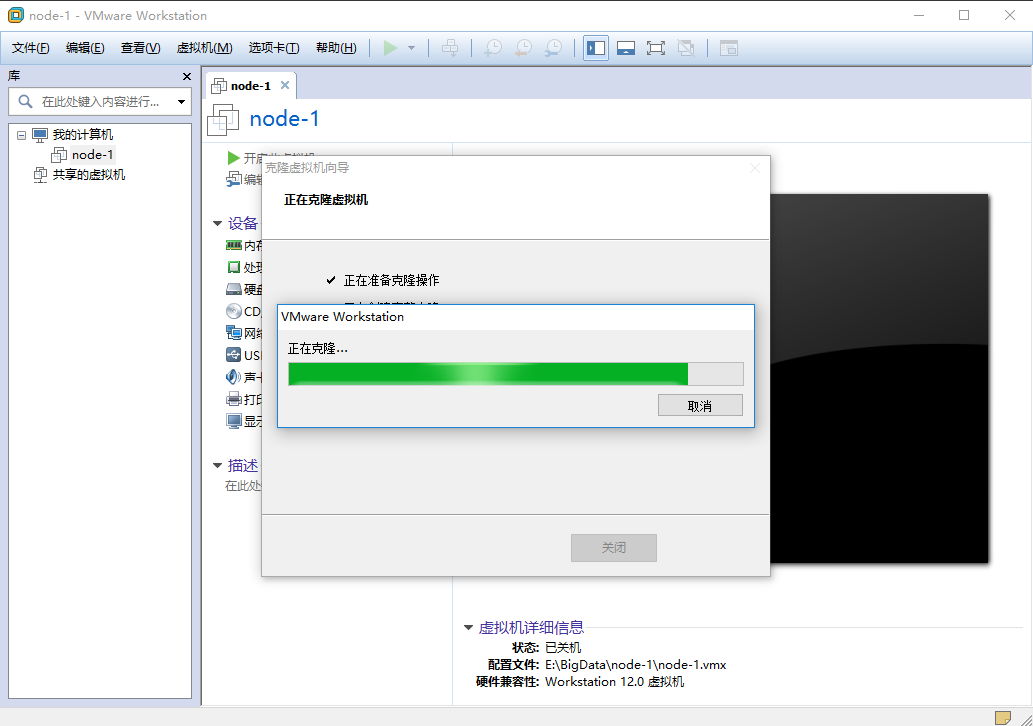


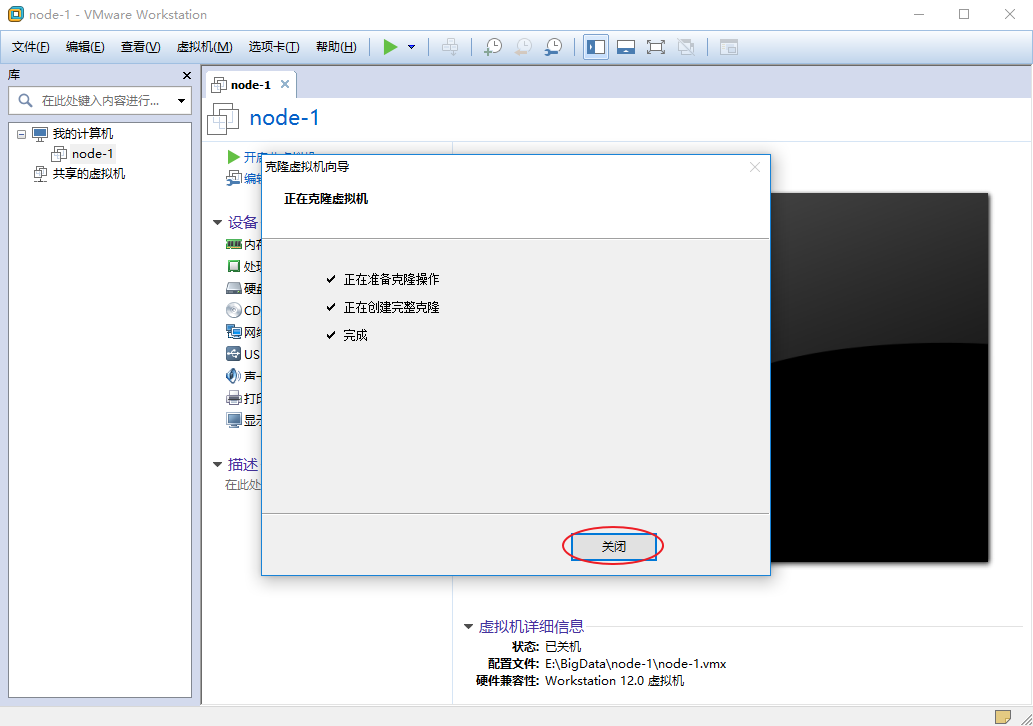




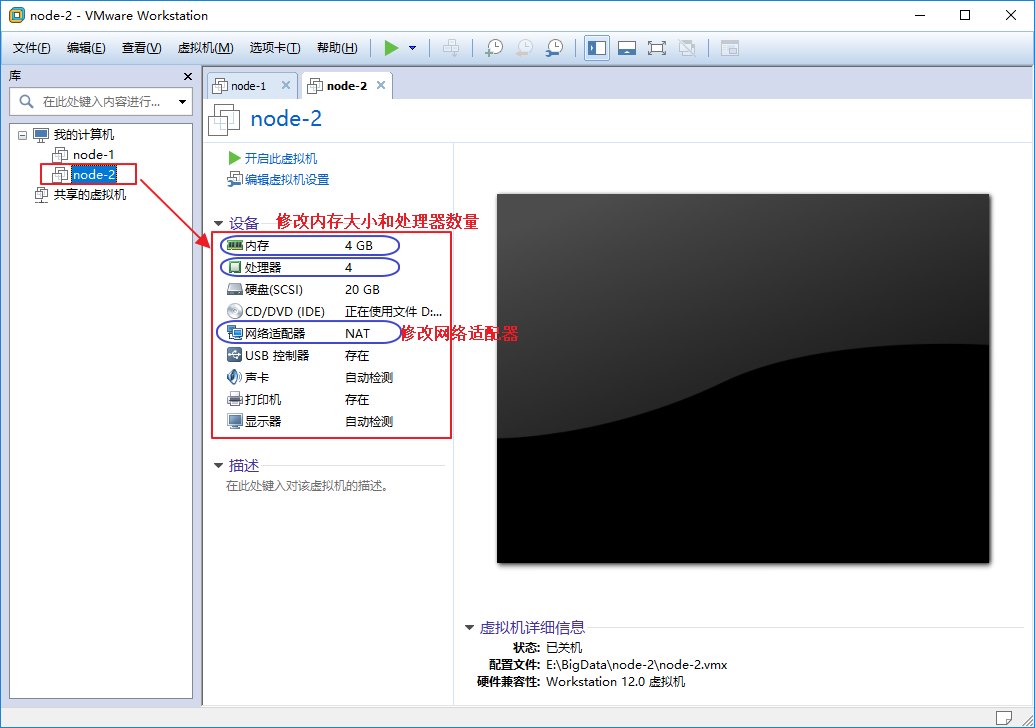


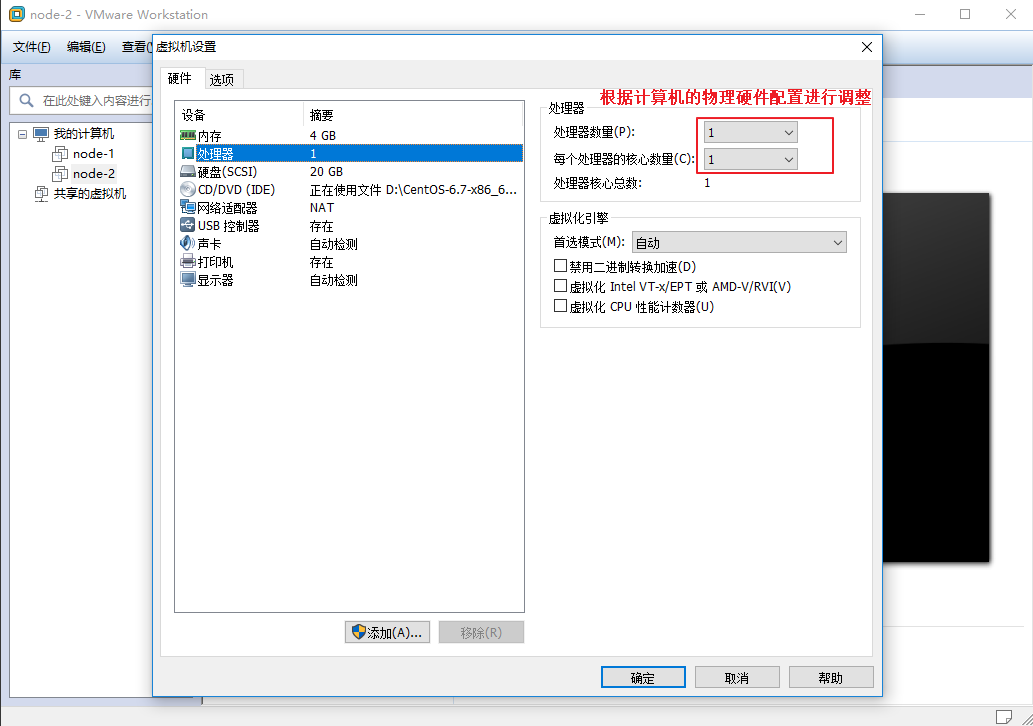


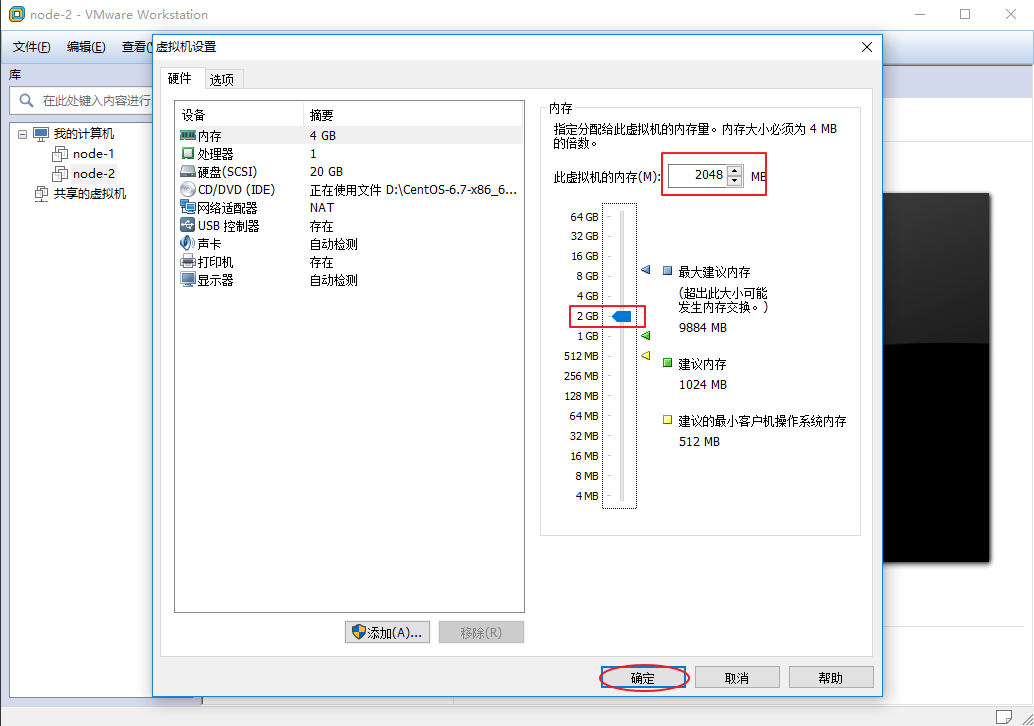


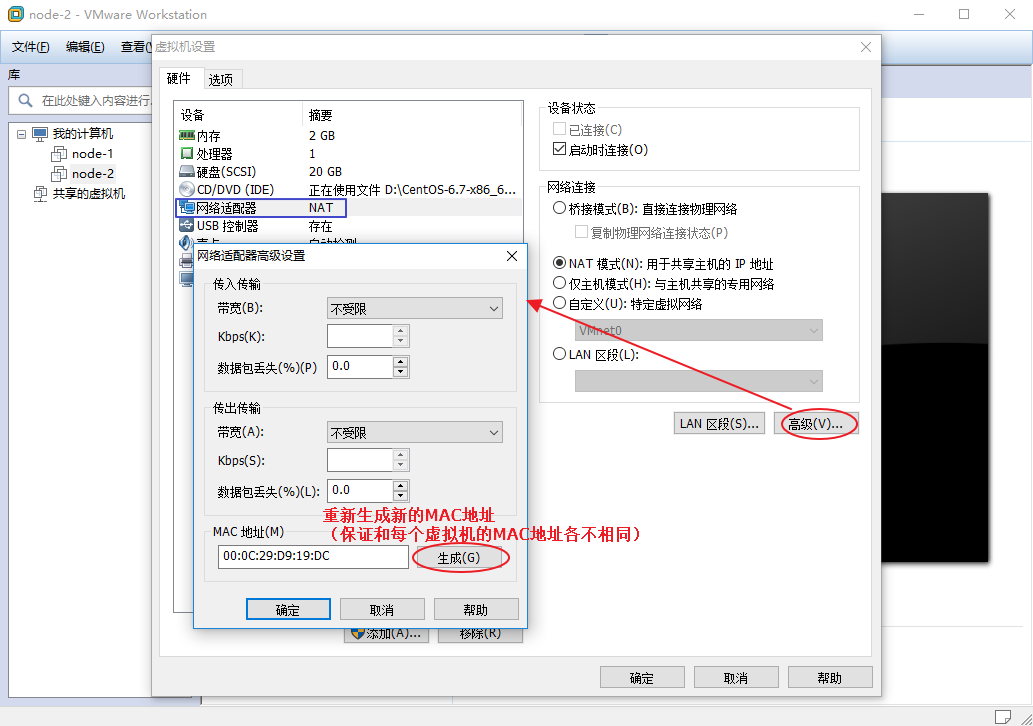


### 1.2、修改虚拟机的硬件配置









### 1.3、修改虚拟机的网络配置

修改主机名hostname:

|  |
| --- |
| vim /etc/hostname  node2.itcast.cn |

修改IP地址

|  |
| --- |
| vim /etc/sysconfig/network-scripts/ifcfg-ens33  TYPE="Ethernet" #网卡类型 以太网  PROXY\_METHOD="none"  BROWSER\_ONLY="no"  BOOTPROTO="none"  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="ens33" #网卡名称  UUID="74c3b442-480d-4885-9ffd-e9f0087c9cf7"  DEVICE="ens33"  ONBOOT="yes" #是否开机启动网卡服务  IPADDR="192.168.88.152" #IP地址  PREFIX="24" #子网掩码 等效: NETMASK=255.255.255.0  GATEWAY="192.168.88.2" #网关服务  DNS1="192.168.88.2" #网关DNS解析  DOMAIN="114.114.114.114" #公网DNS解析 114.114.114.114 谷歌：8.8.8.8 阿里百度DNS  IPV6\_PRIVACY="no |

systemctl restart network # 重启网络服务

## 2. 关闭防火墙

### 2.1、关闭firewalld

|  |
| --- |
| #查看防火墙状态  systemctl status firewalld  #关闭防火墙  systemctl stop firewalld  #关闭防火墙开机自启动  systemctl disable firewalld  #centos服务开启关闭命令  centos6:(某些可以在centos7下使用)  service 服务名 start|stop|status|restart  chkconfig on|off 服务名    centos7:  systemctl start|stop|status|restart 服务名  systemctl disable|enable 服务名 #开机自启动 关闭自启 |

### 2.2、关闭SElinux

安全增强型 Linux（Security-Enhanced Linux）简称 SELinux，它是一个 Linux 内核模块，也是 Linux 的一个安全子系统。

SELinux 主要由美国国家安全局开发。2.6 及以上版本的 Linux 内核都已经集成了 SELinux 模块。

SELinux 主要作用就是最大限度地减小系统中服务进程可访问的资源（最小权限原则）。

注意需要重启才能生效。

|  |
| --- |
| vim /etc/selinux/config  # This file controls the state of SELinux on the system.  # SELINUX= can take one of these three values:  # enforcing - SELinux security policy is enforced.  # permissive - SELinux prints warnings instead of enforcing.  # disabled - No SELinux policy is loaded.  SELINUX=disabled |

## 3. 修改主机名

使用vi编辑器，修改主机名：

vim /etc/hostname

root重启后生效

## 4. 修改hosts映射

分别编辑每台虚拟机的hosts文件，在原有内容的基础上，填下以下内容:

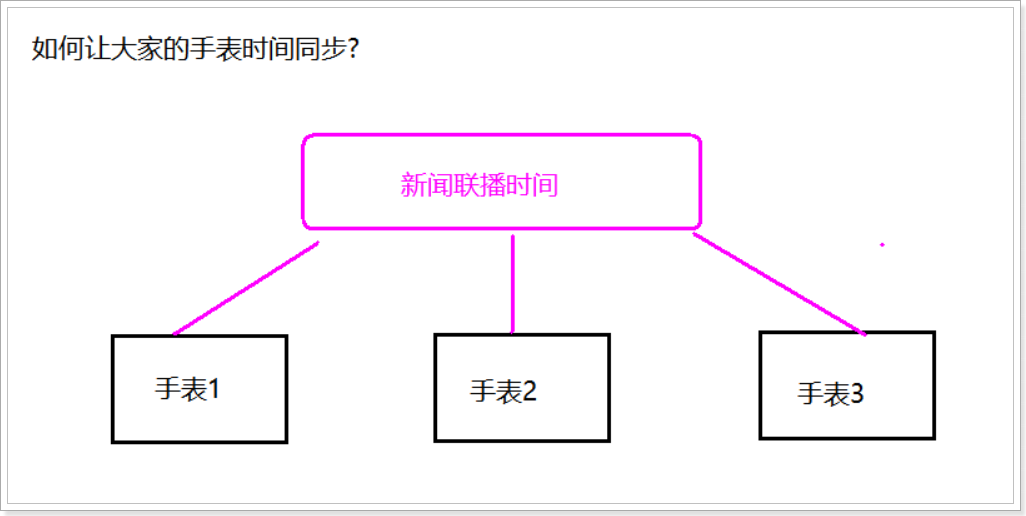
注意：不要修改文件原来的内容，三台虚拟机的配置内容都一样。

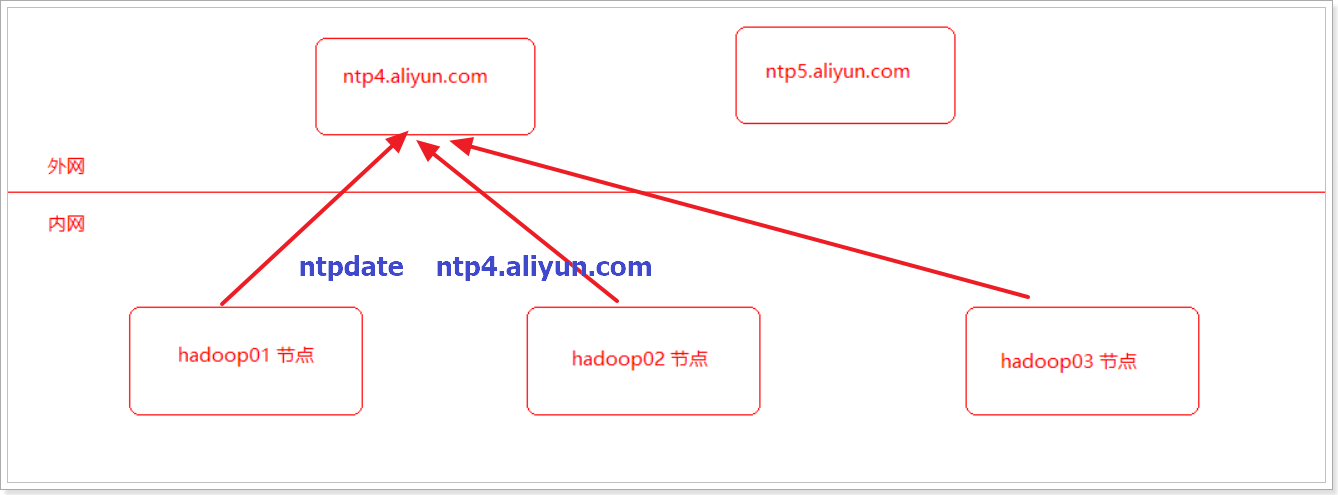
|  |
| --- |
| vim /etc/hosts  127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.88.151 node1.itcast.cn node1  192.168.88.152 node2.itcast.cn node2  192.168.88.153 node3.itcast.cn node3 |

## 5. SSH免密登录

|  |
| --- |
| #实现node1----->node2  #step1  在node1生成公钥私钥  ssh-keygen 一顿回车 在当前用户的home下生成公钥私钥 隐藏文件  [root@node1 .ssh]# pwd  /root/.ssh  [root@node1 .ssh]# ll  total 12  -rw------- 1 root root 1675 May 20 11:59 id\_rsa  -rw-r--r-- 1 root root 402 May 20 11:59 id\_rsa.pub  -rw-r--r-- 1 root root 183 May 20 11:50 known\_hosts  #step2  copy公钥给node2  ssh-copy-id node2  注意第一次需要密码  #step3  [root@node1 .ssh]# ssh node2  Last login: Thu May 20 12:03:30 2021 from node1.itcast.cn  [root@node2 ~]# exit  logout  Connection to node2 closed.  #课程要求：至少打通node1---->node1 node2 node3 这三个免密登录  #至于所有机器之间要不要互相免密登录 看心情 |

## 6. 时钟同步





| **命令** | **英文** | **含义** |
| --- | --- | --- |
| ntpdate 互联网时间服务器地址 | Network Time Protocol | 同步时间 |

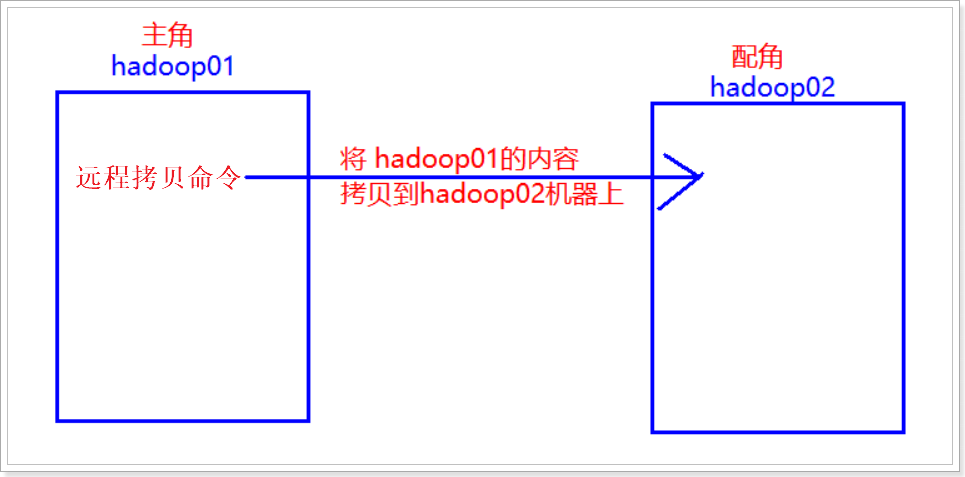
阿里云时钟同步服务器： ntpdate ntp4.aliyun.com

## 7. 远程文件拷贝

远程文件拷贝命令：scp

scp是 remote file copy program 的缩写, scp是远程文件拷贝命令

### 7.1、从本机拷贝到远程服务器



准备工作

# 创建目录

mkdir -p /export/aaa/bbb/ccc

# 创建文件

touch /export/aaa/01.txt /export/aaa/02.txt

# 显示

tree /export/aaa

1、文件

语法格式

# 拷贝文件

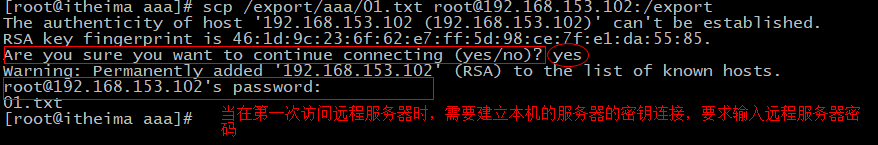
scp local\_file remote\_username@remote\_ip:remote\_folder

需求: 将本地 01.txt 文件 复制到 192.168.153.102 机器的 /export目录下

scp /export/aaa/01.txt root@192.168.153.102:/export

scp /export/aaa/02.txt root@hadoop02:/export

scp /export/aaa/02.txt hadoop02:/export



2、文件夹

语法格式

# 拷贝目录

scp -r local\_folder remote\_username@remote\_ip:remote\_folder

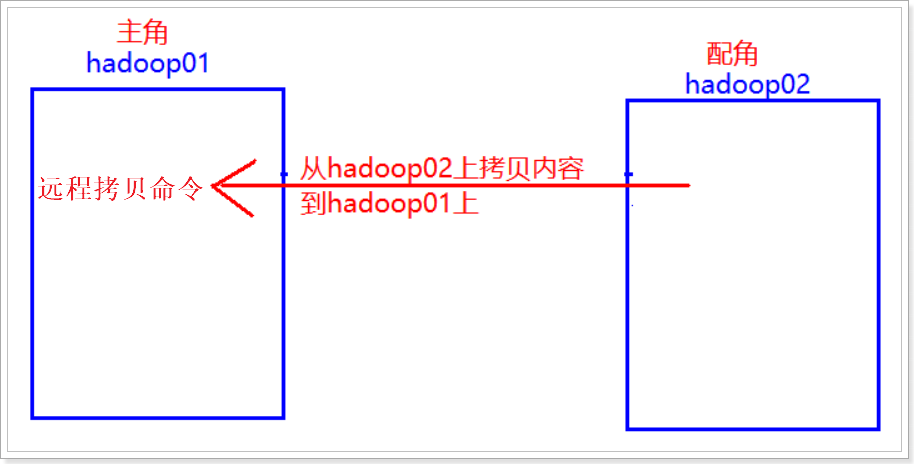
需求: 将本地 aaa 目录 复制到 192.168.150.102 机器的 /export 目录下

scp -r /export/aaa root@192.168.153.102:/export

scp -r /export/aaa root@hadoop02:/export

scp -r /export/aaa hadoop02:/export

### 7.2、从远程服务器复制到本机



准备工作

# 创建目录

mkdir -p /export/a1/b1/c1

# 创建文件

touch /export/a1/111.txt

touch /export/a1/222.txt

# 显示内容

tree /export/a1

1、文件

语法格式

scp remote\_username@remote\_ip:remote\_file local\_folder

练习: 将 hadoop02的 111.txt 文件 复制到 hadoop01的 export目录下

scp root@192.168.100.202:/export/a1/111.txt /export

scp root@hadoop02:/export/a1/222.txt /export

scp hadoop02:/export/a1/222.txt /export

2 目录

语法格式

scp -r remote\_username@remote\_ip:remote\_folder local\_folder

练习: 将hadoop02的 /export/a1/ 目录 复制到 hadoop01 的 export目录下

scp -r root@192.168.100.202:/export/a1 /export

scp -r hadoop02:/export/a1 /export

# 五、Shell编程

## 1. 什么是Shell

### 1.1、简介

Shell 是一个用 C 语言编写的程序， 通过 Shell 用户可以访问操作系统内核服务。

它类似于 DOS 下的 command 和后来的 cmd.exe。 Shell既是一种命令语言，又是一种程序设计语言。

Shell script 是一种为shell编写的脚本程序。 **Shell 编程一般指shell脚本编程**，不是指开发shell自身。

### 1.2、Shell解释器

Shell编程跟java、php编程一样，只要有一个能编写代码的文本编辑器和一个能解释执行的脚本解释器就可以了。

Linux的Shell解释器种类众多，一个系统可以存在多个 shell，可以通过cat /etc/shells 命令查看系统中安装的 shell。



bash 由于易用和免费，在日常工作中被广泛使用。同时， bash 也是大多数Linux 系统默认的 Shell。

总结：java执行需要虚拟机解释器, 同理shell程序执行也需要解析器

## 2. 快速入门

### 2.1、编写Shell脚本

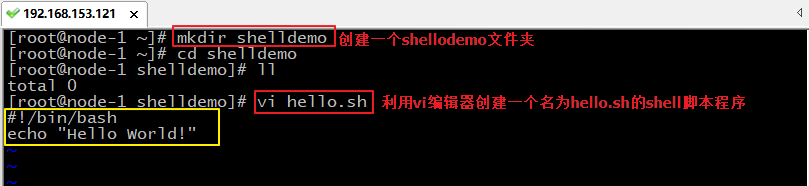
使用 vi 编辑器新建一个文件hello.sh（扩展名并不影响脚本执行）

**#!/bin/bash**

echo "Hello World !"

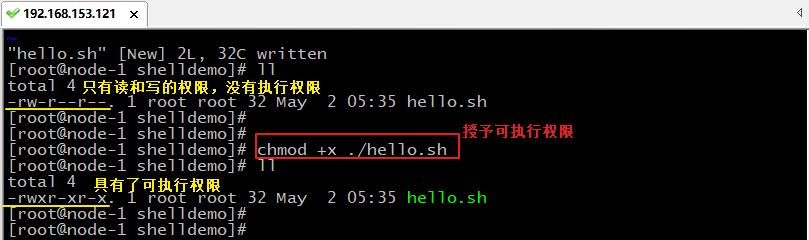
#! 是一个约定的标记，它告诉系统这个脚本需要什么解释器来执行，即使用哪一种 Shell

echo命令 用于向窗口输出文本。



给shell程序赋予执行权限：

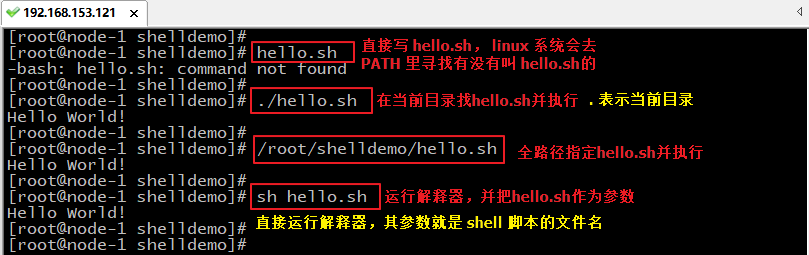
chmod +x ./hello.sh #使脚本具有执行权限



### 2.2、执行Shell脚本

执行shell程序：

./hello.sh #执行脚本



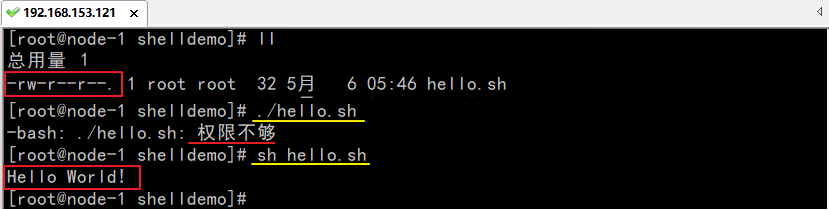
直接写 hello.sh，linux 系统会去 PATH 里寻找有没有叫hello.sh的。

用 ./hello.sh 告诉系统说，就在当前目录找

还可以作为解释器参数运行。 直接运行解释器，其参数就是 shell 脚本的文件名，如：

sh /root/shelldemo/hello.sh

在使用解释器直接执行shell程序这种方式来运行脚本，不需要考虑脚本程序的执行权限了



**小结：**

使用vi编辑器，创建shell程序文件。通常使用.sh作为shell程序后缀名。

shell程序的基本格式：

1、指定解释器信息。 默认：/bin/bash

2、书写shell程序代码

3、保存shell程序代码

4、执行shell程序 提前：给shell程序授予可执行权限

第一种：./xxx.sh #在当前目录中执行shell程序

第二种：/xx/xxx.sh #书写全路径的shell程序

第三种：sh /xx/xxx.sh #把shell程序作用/bin/sh解释器的参数，通过运行解释器来执行shell