

# NSD ADMIN DAY06

1. [案例1：使用Yum软件源](#)
2. [案例3：升级Linux内核](#)
3. [案例3：硬盘分区及格式化](#)
4. [案例4：配置NTP网络时间客户端](#)

## 1 案例1：使用Yum软件源

### 1.1 问题

本例要求为虚拟机 server0指定可用的Yum软件源，相关要求如下：

1. 软件库源为 [http://content.example.com/rhel7.0/x86\\_64/dvd](http://content.example.com/rhel7.0/x86_64/dvd)
2. 将此配置为虚拟机 server0 的默认软件仓库

### 1.2 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：检查现有yum仓库，去除不可用的设置

1) 列出yum库

```
01. [root@server0 ~]# yum repolist
02. 已加载插件：langpacks
03. rhel_dvd | 4.1 kB 00:00:00
04. (1/2): rhel_dvd/group_gz | 134 kB 00:00:00
05. (2/2): rhel_dvd/primary_db | 3.4 MB 00:00:00
06. 源标识          源名称          状态
07. rhel_dvd          Remote classroom copy of dvd      4,305
08. repolist: 4,305
```

2) 移除不可用的yum库配置文件

当执行yum repolist操作报错时，才执行此步骤（否则此步可跳过）。

```
01. [root@server0 ~]# mkdir /etc/yum.repos.d/repobak
02. [root@server0 ~]# mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/repobak/
```

#### 步骤二：添加指定的yum仓库配置

1) 使用yum-config-manager工具建立新配置文件

```
01. [root@server0 ~]# yum-config-manager --add-repo http://content.example.com/rhel7.0/x86
```

```

02. 已加载插件 : langpacks
03. adding repo from: http://content.example.com/rhel7.0/x86_64/dvd
04.
05. [content.example.com_rhel7.0_x86_64_dvd]
06. name=added from: http://content.example.com/rhel7.0/x86_64/dvd
07. baseurl=http://content.example.com/rhel7.0/x86_64/dvd
08. enabled=1

```

## 2) 修改新建的仓库配置，添加gpgcheck=0以禁用GPG签名检查

```

01. [root@server0 ~]# vim /etc/yum.repos.d/content.example.com_rhel7.0_x86_64_dvd
02. [content.example.com_rhel7.0_x86_64_dvd]
03. name=added from: http://content.example.com/rhel7.0/x86_64/dvd
04. baseurl=http://content.example.com/rhel7.0/x86_64/dvd
05. enabled=1
06. gpgcheck=0

```

## 步骤三：确认新配置的yum源可用

```

01. [root@server0 ~]# yum clean all //清理缓存
02. 已加载插件 : langpacks
03. 正在清理软件源 : content.example.com_rhel7.0_x86_64_dvd
04. Cleaning up everything
05. [root@server0 ~]# yum repolist //重新列出可用的源
06. 已加载插件 : langpacks
07. content.example.com_rhel7.0_x86_64_dvd | 4.1 kB 00:00:00
08. (1/2): content.example.com_rhel7.0_x86_64_dvd/group_gz | 134 kB 00:00:00
09. (2/2): content.example.com_rhel7.0_x86_64_dvd/primary_db | 3.4 MB 00:00:00
10. 源标识 源名称 状态
11. content.example.com_rhel7.0_x86_64_dvd added from: http://content.example.com 4,305
12. repolist: 4,305

```

## 2 案例3：升级Linux内核

### 2.1 问题

本例要求为虚拟机 server0安装升级版的新内核：

1. 新版本的内核安装文件可以从以下地址获取：
2. [http://classroom/content/rhel7.0/x86\\_64/errata/Packages/](http://classroom/content/rhel7.0/x86_64/errata/Packages/)

3. 升级内核，并满足下列要求：当系统重新启动后，升级的新内核应该作为默认内核；原来的内核要被保留，并且仍然可以正常启动

## 2.2 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：下载新版内核的安装文件

#### 1) 确认新版内核的下载地址

如果给定的下载地址中未包含kernel-...rpm文件路径，则打开firefox浏览器，访问指定的网址（如图-5所示）。

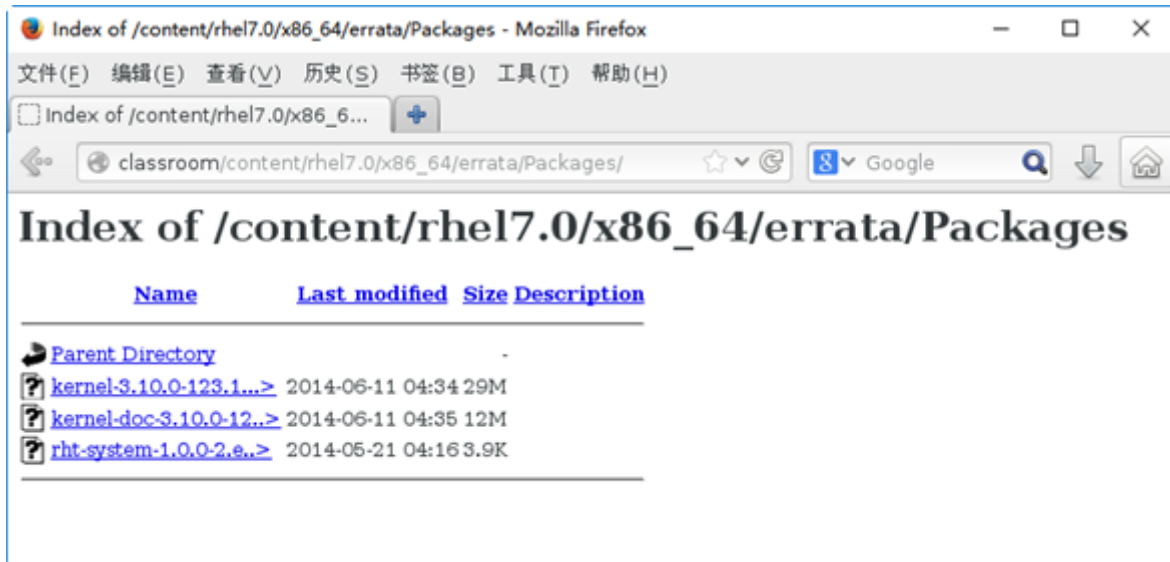


图-5

在打开的网页上找到需要的内核文件，右击对应的链接，选择“Copy Link Location”复制下载地址（如图-6所示）。

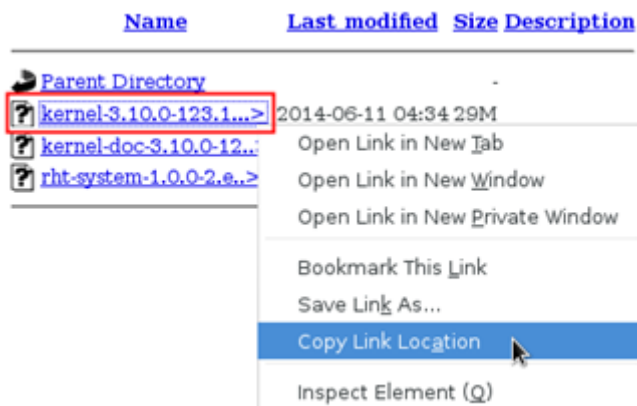


图-6

#### 2) 下载新版内核安装文件

根据前一步获取到的内核下载地址，使用wget命令下载：

01. [root@server0 ~]# wget http://classroom/content/rhel7.0/x86\_64/errata/Packages/kernel-
02. --2016-12-23 22:13:47-- http://classroom/content/rhel7.0/x86\_64/errata/Packages/kern
03. 正在解析主机 classroom (classroom)... 172.25.254.254
04. 正在连接 classroom (classroom)|172.25.254.254|:80... 已连接。

```

05. 已发出 HTTP 请求，正在等待回应... 200 OK
06. 长度：30266784 (29M) [application/x-rpm]
07. 正在保存至：“kernel-3.10.0-123.1.2.el7.x86_64.rpm”
08.
09. 100%[=====>] 30,266,784 40.4MB/s 用E
10.
11. 2016-12-23 22:13:47 (40.4 MB/s) - 已保存 “kernel-3.10.0-123.1.2.el7.x86_64.rpm” [3026
12.
13. [root@server0 ~]# ls -lh kernel-*.rpm //确认下载结果
14. -rw-r--r--. 1 root root 29M 6月 11 2014 kernel-3.10.0-123.1.2.el7.x86_64.rpm

```

## 步骤二：安装新版内核

Linux系统支持安装多个不同版本的内核，开机引导时可以选择使用哪个版本。因此只需要正常安装新版内核即可。

### 1) 查看现有内核版本

```

01. [root@server0 ~]# uname -r
02. 3.10.0-123.el7.x86_64

```

### 2) 安装新版本内核

```

01. [root@server0 ~]# rpm -ivh kernel-3.10.0-123.1.2.el7.x86_64.rpm
02. 警告：kernel-3.10.0-123.1.2.el7.x86_64.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID fd431c
03. 准备中... ##### [100%]
04. 正在升级/安装...
05. 1:kernel-3.10.0-123.1.2.el7 ##### [100%]
06. //此处需耐心等待，千万别强行终止
07. [root@server0 ~]#

```

## 步骤三：确认新内核版本

### 1) 重启系统

```

01. [root@server0 ~]# reboot
02. ...

```

### 2) 登入系统，确认使用的内核已是新版本

01. [root@server0 ~]# uname -r
02. 3.10.0-123.1.2.el7.x86\_64

## 3 案例3：硬盘分区及格式化

### 3.1 问题

本例要求熟悉硬盘分区结构，使用fdisk分区工具在磁盘 /dev/vdb 上按以下要求建立分区：

1. 采用默认的 msdos 分区模式
2. 第1个分区 /dev/vdb1 的大小为 200MiB
3. 第2个分区 /dev/vdb2 的大小为 2000MiB
4. 第3个分区 /dev/vdb3 的大小为 1000MiB

完成分区后，能够配置开机自动挂载 /dev/vdb2 分区：

1. 文件系统类型为 EXT4
2. 将其挂载到 /mnt/part2 目录

### 3.2 方案

fdisk分区工具用来建立msdos分区方案，其交互模式中的主要指令如下：

- m：列出指令帮助
- p：查看当前的分区表信息
- n：新建分区
- d：删除分区
- t：更改分区标识
- q：放弃分区更改并退出
- w：保存对分区表所做的更改

### 3.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：新建分区表

- 1) 打开fdisk工具，操作磁盘/dev/vdb

01. [root@server0 ~]# fdisk /dev/vdb
02. Welcome to fdisk (util-linux 2.23.2).
- 03.
04. Changes will remain in memory only, until you decide to write them.
05. Be careful before using the write command.
- 06.
07. Device does not contain a recognized partition table
08. Building a new DOS disklabel with disk identifier 0x9ac1bc10.
- 09.
10. Command (m for help): //交互操作提示信息

## 2) 新建第1个分区/dev/vdb1

```

01. Command (m for help): n //新建分区
02. Partition type:
03.   p primary (0 primary, 0 extended, 4 free)
04.   e extended
05. Select (default p): p //类型为p (主分区)
06. Partition number (1-4, default 1): 1 //分区编号1
07. First sector (2048-20971519, default 2048): //起始位置默认
08. Using default value 2048
09. Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519): +200M
10. Partition 1 of type Linux and of size 200 MiB is set //结束位置+200MiB大小
11.
12. Command (m for help): p //确认当前分区表
13. .. ..
14.   Device Boot      Start         End      Blocks   Id  System
15.  /dev/vdb1          2048       411647      204800    83  Linux

```

## 3) 新建第2个分区/dev/vdb2

```

01. Command (m for help): n
02. Partition type:
03.   p primary (1 primary, 0 extended, 3 free)
04.   e extended
05. Select (default p): p //类型为p (主分区)
06. Partition number (2-4, default 2): 2 //分区编号2
07. First sector (411648-20971519, default 411648): //起始位置默认
08. Using default value 411648
09. Last sector, +sectors or +size{K,M,G} (411648-20971519, default 20971519): +2000M
10. Partition 2 of type Linux and of size 2 GiB is set //结束位置+2000MiB大小
11.
12. Command (m for help): p //确认当前分区表
13. .. ..
14.   Device Boot      Start         End      Blocks   Id  System
15.  /dev/vdb1          2048       411647      204800    83  Linux
16.  /dev/vdb2        411648      4507647     2048000    83  Linux

```

## 4) 新建第3个分区/dev/vdb3

```

01. Command (m for help): n
02. Partition type:
03.   p primary (2 primary, 0 extended, 2 free)
04.   e extended
05. Select (default p): p
06. Partition number (3,4, default 3): 3
07. First sector (4507648-20971519, default 4507648):
08. Using default value 4507648
09. Last sector, +sectors or +size{K,M,G} (4507648-20971519, default 20971519): +1000M
10. Partition 3 of type Linux and of size 1000 MiB is set
11.
12. Command (m for help): p //确认当前分区表
13. .. ..
14.   Device Boot      Start         End      Blocks   Id  System
15.   /dev/vdb1        2048         411647     204800    83  Linux
16.   /dev/vdb2        411648        4507647     2048000    83  Linux
17.   /dev/vdb3        4507648        6555647     1024000    83  Linux

```

## 5) 调整分区类型标识 (可选)

将/dev/vdb1的类型 (默认为83,表示EXT2/3/4分区) 修改为8e (LVM设备) :

```

01. Command (m for help): t //修改分区类型标识
02. Partition number (1-3, default 3): 1 //指定第1个分区
03. Hex code (type L to list all codes): 8e //类型改为8e
04. Changed type of partition 'Linux' to 'Linux LVM'
05.
06. Command (m for help): p //确认当前分区表
07. .. ..
08.   Device Boot      Start         End      Blocks   Id  System
09.   /dev/vdb1        2048         411647     204800    8e  Linux LVM
10.   /dev/vdb2        411648        4507647     2048000    83  Linux
11.   /dev/vdb3        4507648        6555647     1024000    83  Linux

```

## 6) 保存分区更改, 退出fdisk分区工具

```

01. Command (m for help): w //保存并退出

```

02. The partition table has been altered!
- 03.
04. Calling `ioctl()` to re-read partition table.
05. Syncing disks.

## 6) 刷新分区表

01. `[root@server0 ~]# partprobe /dev/vdb` //重新检测磁盘分区
02. //或者
03. `[root@server0 ~]# reboot` //对已使用中磁盘的分区调整，应该重启一次
04. ...

## 步骤二：格式化及挂载分区

### 1) 将分区/dev/vdb2格式化为EXT4文件系统

01. `[root@server0 ~]# mkfs.ext4 /dev/vdb2`
02. ...
03. Allocating group tables: done
04. Writing inode tables: done
05. Creating journal (8192 blocks): done
06. Writing superblocks and filesystem accounting information: done

### 2) 配置开机自动挂载

01. `[root@server0 ~]# vim /etc/fstab`
02. ...
03. `/dev/vdb2 /mnt/part2 ext4 defaults 0 0`

### 3) 创建挂载点，并验证挂载配置

01. `[root@server0 ~]# mkdir /mnt/part2` //创建挂载点
02. `[root@server0 ~]# mount -a` //挂载fstab中的可用设备
03. `[root@server0 ~]# df -hT /mnt/part2/` //检查文档所在的文件系统及设备
04. Filesystem Type Size Used Avail Use% Mounted on
05. `/dev/vdb2 ext4 1.9G 5.9M 1.8G 1% /mnt/part2`



## 4 案例4：配置NTP网络时间客户端

### 4.1 问题

本例要求配置虚拟机 server0，能够自动校对系统时间。相关信息如下：

1. NTP服务器位于 classroom.example.com
2. 此客户端的系统时间应当与NTP服务器的时间保持同步

### 4.2 方案

NTP服务端可以为客户端提供标准的日期时间。

在RHEL7主机中可以配置软件包chrony来使用NTP时间同步。

### 4.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：确认已安装NTP客户端软件包chrony

通常系统会默认安装此软件包：

01. [root@server0 ~]# rpm -q chrony
02. chrony-1.29.1-1.el7.x86\_64

如果检查发现此软件包没有安装，请通过yum命令安装：

01. [root@server0 ~]# yum -y install chrony
02. .. ..

如果发现此软件包已经安装，但配置文件/etc/chrony.conf损坏或信息丢失，可删除此文件后重新安装chrony软件包：

01. [root@server0 ~]# rm -rf /etc/chrony.conf
02. [root@server0 ~]# yum -y reinstall chrony
03. 已加载插件：langpacks
04. 正在解决依赖关系
05. --> 正在检查事务
06. ---> 软件包 chrony.x86\_64.0.1.29.1-1.el7 将被 重新安装
07. --> 解决依赖关系完成
08. .. ..

#### 步骤二：调整NTP客户端配置

- 1) 修改/etc/chrony.conf文件内的server配置

移除不可用的NTP服务器记录，正确添加可用的记录：

```
01. [root@server0 ~]# vim /etc/chrony.conf
02. .. ..
03. #server 0.rhel.pool.ntp.org iburst //注释掉默认的server配置，
04. #server 1.rhel.pool.ntp.org iburst
05. #server 2.rhel.pool.ntp.org iburst
06. #server 3.rhel.pool.ntp.org iburst
07. server classroom.example.com iburst //添加新的配置
08. .. ..
```

## 2) 开启NTP时间同步

```
01. [root@server0 ~]# timedatectl //查看现有状态
02. .. ..
03. NTP enabled: no //NTP可能尚未启用
04. NTP synchronized: no //尚未完成过一次NTP同步
05. .. ..
06. [root@server0 ~]# timedatectl set-ntp yes //启用NTP同步
07. [root@server0 ~]# timedatectl //查看启用后的状态
08. .. ..
09. NTP enabled: yes //NTP已经启用
10. NTP synchronized: no //尚未完成过一次NTP同步
11. .. ..
```

## 步骤三：测试NTP时间同步

### 1) 先设置一个错误的系统日期时间

```
01. [root@server0 ~]# date -s '2001-09-11 11:30:00' //调整日期时间
02. 2001年 09月 11日 星期二 11:30:00 CST
03. [root@server0 ~]# date //确认调整结果
04. 2001年 09月 11日 星期二 11:30:01 CST
```

### 2) 启动系统服务chronyd，并设为开机自启

```
01. [root@server0 ~]# systemctl restart chronyd
02. [root@server0 ~]# systemctl enable chronyd
```

### 3) 重新查看当前的系统时间

重启chronyd服务后稍等片刻，当前系统的日期时间应该恢复正常（与NTP服务器保持一致）：

01. [root@server0 ~]# date
02. 2016年 12月 23日 星期五 23:44:53 CST

再次执行timedatectl查看，会发现NTP synchronized的值已经变成yes：

01. [root@server0 ~]# timedatectl
02. .. ..
03. NTP enabled: yes //NTP已经启用
04. NTP synchronized: no //已经完成过一次NTP同步
05. .. ..