

# 云计算应用管理

**NSD ENGINEER**

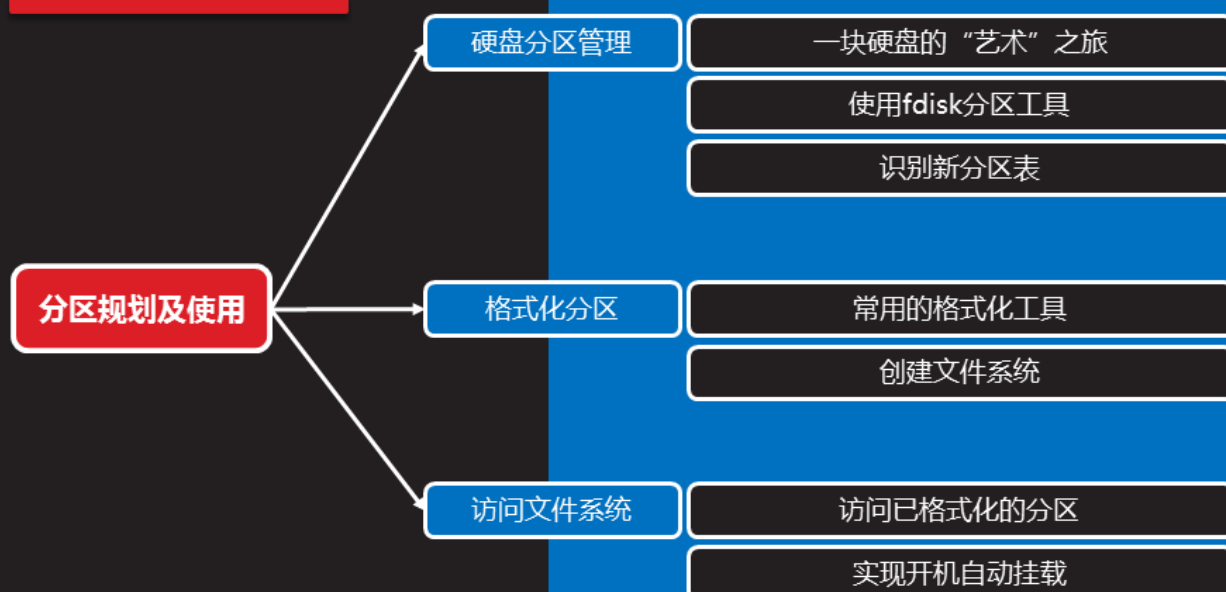
**DAY01**

# 内容

上午	09:00 ~ 09:30	分区规划及使用
	09:30 ~ 10:20	
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	LVM逻辑卷
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



## 分区规划及使用



# 硬盘分区管理

## 一块硬盘的“艺术”之旅

- 识别硬盘 => 分区规划 => 格式化 => 挂载使用



知识讲解

- **MBR/msdos** 分区模式
  - 1~4个主分区，或者 0~3个主分区+1个扩展分区（n个逻辑分区）
  - 最大支持容量为 2.2TB 的磁盘
  - 扩展分区不能格式化



# 使用fdisk分区工具

## 知识讲解

- 查看分区表
  - `fdisk -l /dev/sda`
- 修改硬盘的分区表
  - `fdisk 硬盘设备`

```
[root@server0 ~]# fdisk /dev/vdb
```

```
.. ..
```

```
Command (m for help): p
```

```
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Device Boot      Start         End      Blocks   Id  System
/dev/vdb1        2048        411647        204800   83   Linux
/dev/vdb2       411648       4507647       2048000   83   Linux
/dev/vdb3       4507648       6555647       1024000   83   Linux
```

常用交互指令：

m	列出指令帮助
p	查看现有的分区表
n	新建分区
d	删除分区
q	放弃更改并退出
w	保存更改并退出



# 识别新分区表

## 知识讲解

- 使用 `partprobe` 命令
  - 当硬盘的分区表被更改以后，需要将分区表的变化及时通知 Linux 内核，最好 `reboot` 一次
  - 否则在访问分区时可能会找不到准确的设备

```
[root@server0 ~]# partprobe /dev/vdb
```

```
// 或者
```

```
[root@server0 ~]# reboot
```



# 格式化分区

## 常用的格式化工具

- mkfs 工具集
  - mkfs.ext3 分区设备路径
  - mkfs.ext4 分区设备路径
  - mkfs.xfs 分区设备路径
  - mkfs.vfat -F 32 分区设备路径

[root@server0 ~]# mkfs.<TAB><TAB>

mkfs.btrfs

mkfs.ext3

mkfs.minix

mkfs.xfs

mkfs.cramfs

mkfs.ext4

mkfs.msdos

mkfs.ext2

mkfs.fat

mkfs.vfat

# 创建文件系统

知识讲解

- 按照不同的类型格式化分区

- /dev/vdb2 ==> EXT4 类型

- /dev/vdb3 ==> XFS 类型

```
[root@server0 ~]# mkfs.ext4 /dev/vdb2
```

```
.. ..
```

```
[root@server0 ~]# mkfs.xfs /dev/vdb3
```

```
.. ..
```



# 访问文件系统

## 访问已格式化的分区

知识讲解

- 使用 mount 命令挂载，并访问测试

```
[root@server0 ~]# mkdir /mnt/part2
[root@server0 ~]# mount /dev/vdb2 /mnt/part2

[root@server0 ~]# cp /etc/fstab /mnt/part2/fstab.txt
[root@server0 ~]# ls /mnt/part2
fstab.txt lost+found
```

- 使用 df 检查使用情况

```
[root@server0 ~]# df -hT /mnt/part2
Filesystem      Type  Size  Used Avail  Use% Mounted on
/dev/vdb2       ext4  1.9G   5.9M  1.8G   1%   /mnt/part2
```



## 实现开机自动挂载

知识讲解

- 配置文件 /etc/fstab 的记录格式

– 设备路径 挂载点 类型 参数 备份标记 检测顺序

```
[root@server0 ~]# mkdir /mnt/part3 //创建挂载点
[root@server0 ~]# vim /etc/fstab

.. ..
UUID=9bf6b9f.. .. / xfs defaults 1 1
/dev/vdb3 /mnt/part3 xfs defaults 0 0
```

```
[root@server0 ~]# mount -a //检测配置
[root@server0 ~]# df -hT /mnt/part3
Filesystem      Type  Size  Used Avail  Use% Mounted on
/dev/vdb3       xfs   997M  33M   965M   4%   /mnt/part3
```



# 案例1：硬盘分区及格式化

## 课堂练习

1. 在磁盘 /dev/vdb 上按以下要求建立分区：

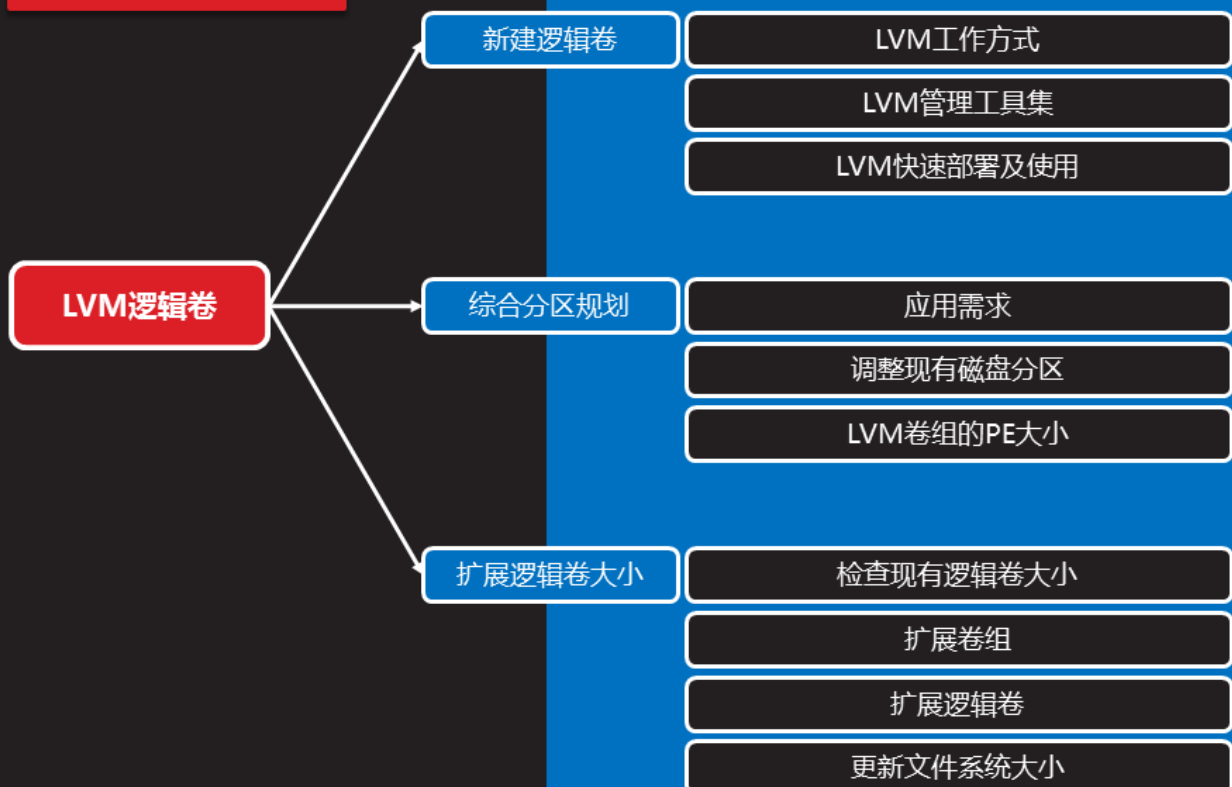
- 1) 采用默认的 msdos 分区模式
- 2) 第1个分区 /dev/vdb1 的大小为 200MiB
- 3) 第2个分区 /dev/vdb2 的大小为 2000MiB
- 4) 第3个分区 /dev/vdb3 的大小为 1000MiB

2. 实现开机自动挂载 /dev/vdb2 分区

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



## LVM逻辑卷



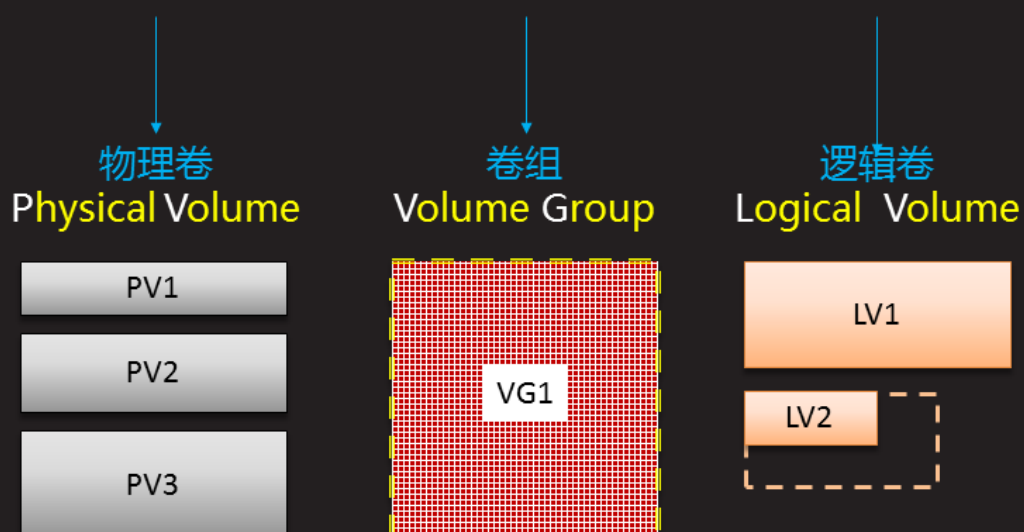


# 新建逻辑卷

## LVM工作方式

- 在“分区 --> 格式化”中间增加的一个逻辑层
  - 零散空闲存储 ---- 整合的虚拟磁盘 ---- 虚拟的分区

知识讲解



# LVM管理工具集

知识讲解

功能	物理卷管理	卷组管理	逻辑卷管理
Scan 扫描	pvscan	vgscan	lvscan
Create 创建	pvcreate	vgcreate	lvcreate
Display 显示	pvdisk	vgdisplay	lvdisplay
Remove 删除	pvremove	vgremove	lvremove
Extend 扩展	/	vgextend	lvextend



## LVM快速部署及使用

知识讲解

- 基本思路
  - 准备至少一个空闲分区 ( /dev/vdb1 )
  - 创建卷组 : **vgcreate** 卷组名 空闲分区...
  - 创建逻辑卷 : **lvcreate** -L 大小 -n 名称 卷组名

```
[root@server0 ~]# vgcreate systemvg /dev/vdb1
```

```
.. ..
```

```
[root@server0 ~]# lvcreate -L 180M -n vo systemvg
Logical volume "vo" created
```

```
[root@server0 ~]# lvscan
```

```
ACTIVE          '/dev/systemvg/vo' [180.00 MiB] inherit
```



## 案例2：新建一个逻辑卷

使用分区 /dev/vdb1 构建 LVM 存储

- 1) 新建一个名为 systemvg 的卷组
- 2) 在此卷组中创建名为 vo 的逻辑卷，大小180MiB
- 3) 将逻辑卷 vo 格式化为 EXT4 文件系统
- 4) 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为 "I AM KING."

课堂练习



## 综合分区规划

## 应用需求

知识讲解

- 硬盘 /dev/vdb 剩余空间的分区调整
  - MBR分区模式，已有三个主分区 /dev/vdb[1-3]
  - 需要新增三个分区 /dev/vdb[5-7]，  
分区大小依次为 500MiB、2000MiB、512MiB
  - 分别用于后续的 逻辑卷扩展、新逻辑卷、交换分区



## 调整现有磁盘分区

知识讲解

- 需要注意的事项
  - 分区数量  $3+3>4$ ，需要建扩展分区
  - 将所有的剩余空间分配给新建的扩展分区
  - 从扩展分区中再新建3个逻辑分区
  - 不要动原有分区、保存更改后 **reboot** 以刷新分区表



## LVM卷组的PE大小

### 知识讲解

- 创建卷组的时候设置PE大小

– `vgcreate -s PE大小 卷组名 空闲分区...`

```
[root@server0 ~]# vgcreate -s 16MiB datastore /dev/vdb6
Physical volume "/dev/vdb6" successfully created
Volume group "datastore" successfully created
```

- 创建逻辑卷的时候指定PE个数

– `lvcreate -l PE个数 -n 逻辑卷名 卷组名`

```
[root@server0 ~]# lvcreate -l 50 -n database datastore
Logical volume "database" created
[root@server0 ~]# lvscan
ACTIVE          '/dev/datastore/database' [800.00 MiB] inherit
ACTIVE          '/dev/systemvg/vo' [180.00 MiB] inherit
```



## 案例3：调整现有磁盘的分区

### 课堂练习

### 1. 修改 /dev/vdb 的分区表

- 不更改原有分区，利用剩余空间新增三个分区，大小依次为：500MiB、2000MiB、512MiB

### 2. 基于刚建立的 2000MiB 分区构建新的 LVM 存储

- 新的逻辑卷命名为 `database`，大小为50个物理扩展单元（Physical Extent），属于 `datastore` 卷组
- 在 `datastore` 卷组中的所有逻辑卷，其物理扩展单元（Physical Extent）的大小为16MiB
- 使用 EXT3 文件系统对逻辑卷 `database` 格式化，此逻辑卷应该在开机时自动挂载到 `/mnt/database` 目录



# 扩展逻辑卷大小

## 检查现有逻辑卷大小

- 使用 `lvscan` 命令，找到逻辑卷的大小、卷组名

```
[root@server0 ~]# lvscan | grep vo
ACTIVE          '/dev/systemvg/vo' [180.00 MiB] inherit
.. ..
```

- 检查所在卷组的剩余空间，看是否满足扩展的需要

```
[root@server0 ~]# vgdisplay systemvg | grep Free
Free PE / Size    4 / 16.00 MiB
```

## 扩展卷组

知识讲解

- 当卷组的剩余空间不足时，需要先扩展卷组
  - 否则扩展逻辑卷时会报错 “Insufficient free space”
  - 扩展卷组：  
`vgextend 卷组名 空闲分区...`

```
[root@server0 ~]# vgextend systemvg /dev/vdb5
Physical volume "/dev/vdb5" successfully created
Volume group "systemvg" successfully extended
```

```
[root@server0 ~]# vgdisplay systemvg | grep 'Free'
Free PE / Size      128 / 512.00 MiB
```



## 扩展逻辑卷

知识讲解

- 当卷组的剩余空间充足时，可直接扩展逻辑卷
  - 扩展逻辑卷：

`lvextend -L 新大小 /dev/卷组名/逻辑卷名`

```
[root@server0 ~]# lvextend -L 300MiB /dev/systemvg/vo
Extending logical volume vo to 300.00 MiB
Logical volume vo successfully resized
```

```
[root@server0 ~]# lvscan
```

```
.. ..
ACTIVE          '/dev/systemvg/vo' [300.00 MiB] inherit
```



## 更新文件系统大小

知识讲解

- 对于格式化过的逻辑卷，扩容后应该通知Linux内核
  - 否则，使用 df 等工具时看到的仍是旧大小
  - 刷新文件系统容量：

`resize2fs` EXT3/4设备 或者 `xfs_growfs` XFS设备

```
[root@server0 ~]# df -hT /vo
Filesystem                Type      Size  Used Avail Use% Mounted on
/dev/mapper/systemvg-vo ext4    171M   1.6M  157M   1% /vo
```

```
[root@server0 ~]# resize2fs /dev/systemvg/vo
```

```
.. ..
```

```
[root@server0 ~]# df -hT /vo
Filesystem                Type      Size  Used Avail Use% Mounted on
/dev/mapper/systemvg-vo ext4    287M   2.1M  268M   1% /vo
```



## 案例4：扩展逻辑卷的大小

将逻辑卷 vo 的大小调整为 300MiB

- 1) 原文件系统中的内容必须保持完整
- 2) 必要时可使用之前准备的分区 /dev/vdb5 来补充空间
- 3) 注意：分区大小很少能完全符合要求的大小，所以大小在 270MiB 和 300MiB 之间都是可以接受的

课堂练习





# 总结和答疑



# 逻辑卷扩展

## 问题现象

- 利用lvextend命令
  - 逻辑卷扩展后，df命令信息未变化

知识讲解



## 故障分析及排除

- 原因分析
  - 没刷新文件系统
- 解决办法
  - 刷新ext4文件系统：resize2fs
  - 刷新xfs文件系统：xfs\_growfs

知识讲解



