

章 7

管理逻辑卷

目标

使用命令行创建和管理含有文件系统和交换空间的逻辑卷。

培训目标

- 从存储设备创建和管理逻辑卷，将其格式化为相应的文件系统，或为其准备交换空间。
- 文件系统添加和删除分配给卷组的存储，并以非破坏性方式扩展已格式化为文件系统的逻辑卷的大小。

章节

- 创建逻辑卷（及引导式练习）
- 扩展逻辑卷（及引导式练习）

实验

管理逻辑卷

创建逻辑卷

培训目标

学完本节后，您应能够：

- 描述逻辑卷管理的组件和概念。
- 实施 LVM 存储。
- 显示 LVM 组件信息。

逻辑卷管理 (LVM) 概念

逻辑卷和逻辑卷管理有助于更加轻松地管理磁盘空间。如果托管逻辑卷的文件系统需要更多空间，可以将其卷组中的可用空间分配给逻辑卷，并且可以调整文件系统的大小。如果磁盘开始出现错误，可以将替换磁盘注册为物理卷放入卷组中，并且逻辑卷的区块可迁移到新磁盘。

LVM 定义

物理设备

物理设备是用于保存逻辑卷中所存储数据的存储设备。它们是块设备，可以是磁盘分区、整个磁盘、RAID 阵列或 SAN 磁盘。设备必须初始化为 LVM 物理卷，才能与 LVM 结合使用。整个设备将用作一个物理卷。

物理卷 (PV)

在 LVM 系统中使用设备之前，必须将设备初始化为物理卷。LVM 工具会将物理卷划分为物理区块 (PE)，它们是充当物理卷上最小存储块的小块数据。

卷组 (VG)

卷组是存储池，由一个或多个物理卷组成。它在功能上与基本存储中的整个磁盘相当。一个 PV 只能分配给一个 VG。VG 可以包含未使用的空间和任意数目的逻辑卷。

逻辑卷 (LV)

逻辑卷根据卷组中的空闲物理区块创建，提供应用、用户和操作系统所使用的“存储”设备。LV 是逻辑区块 (LE) 的集合，LE 映射到物理区块 (PV 的最小存储块)。默认情况下，每个 LE 将映射到一个 PE。设置特定 LV 选项将会更改此映射；例如，镜像会导致每个 LE 映射到两个 PE。

实施 LVM 存储

创建 LVM 存储需要几个步骤。第一步是确定要使用的物理设备。在组装完一组合适的设备之后，系统会将它们初始化为物理卷，以便将它们识别为属于 LVM。这些物理卷随即被合并到卷组中。此时将会创建一个磁盘空间池，从中可以分配逻辑卷。利用卷组的可用空间创建的逻辑卷可以格式化为文件系统、作为交换空间激活，也可以实现持久挂载或激活。

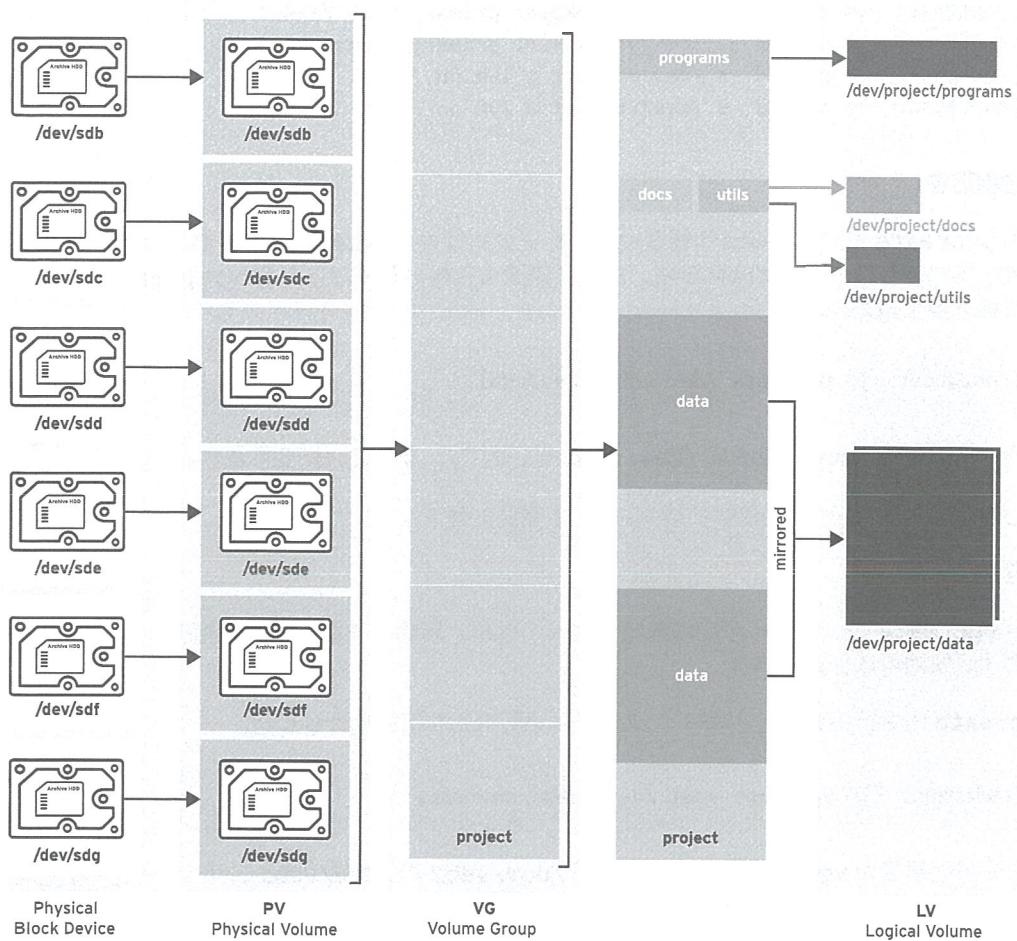


图 7.1: 逻辑卷管理组件

LVM 提供了一组全面的命令行工具，用于实施和管理 LVM 存储。这些命令行工具可用在脚本中，从而使它们更适于自动化。



重要

以下示例使用设备 **vdb** 及其分区来说明 LVM 命令。实际上，这些示例需要为系统所使用的磁盘和磁盘分区使用正确的设备。使用 **lsblk**、**blkid** 或 **cat /proc/partitions** 命令标识系统上的设备。

创建逻辑卷

要创建逻辑卷，请执行以下步骤：

准备物理设备。

使用 **parted**、**gdisk** 或 **fdisk** 创建新分区，以便与 LVM 结合使用。在 LVM 分区上，始终将分区类型设置为 **Linux LVM**；对于 MBR 分区，使用 **0x8e**。如有必要，使用 **partprobe** 向内核注册新分区。

也可以使用完整磁盘、RAID 阵列或 SAN 磁盘。

只有当没有已准备好的物理设备并且需要新物理卷来创建或扩展卷组时，才需要准备物理设备。

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1MiB 769MiB
[root@host ~]# parted -s /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted -s /dev/vdb set 1 lvm on
[root@host ~]# parted -s /dev/vdb set 2 lvm on
```

创建物理卷。

使用 **pvccreate** 将分区（或其他物理设备）标记为物理卷。**pvccreate** 命令会将物理卷分成若干固定大小的物理区块 (PE)，如 4 MiB 块。您可以使用以空格分隔的设备名称作为 **pvccreate** 的参数，同时标记多个设备。

```
[root@host ~]# pvccreate /dev/vdb2 /dev/vdb1
```

此命令会将设备 **/dev/vdb2** 和 **/dev/vdb1** 标记为 PV，准备好分配到卷组中。

仅当没有空闲的 PV 可以创建或扩展 VG 时，才需要创建 PV。

创建卷组。

使用 **vgcreate** 将一个或多个物理卷结合为一个卷组。卷组在功能上与硬盘相当；利用卷组中的可用物理区块池可以创建逻辑卷。

vgcreate 命令行由卷组名后跟一个或多个要分配给此卷组的物理卷组成。

```
[root@host ~]# vgcreate vg01 /dev/vdb2 /dev/vdb1
```

此命令将创建名为 **vg01** 的 VG，它的大小是 **/dev/vdb2** 和 **/dev/vdb1** 这两个 PV 的大小之和（以 PE 单位计）。

仅当 VG 尚不存在时，才需要创建 VG。可能会出于管理原因创建额外的 VG，用于管理 PV 和 LV 的使用。否则，可在需要时扩展现有 VG 以容纳新的 LV。

创建逻辑卷。

使用 **lvcreate** 可根据卷组中的可用物理区块创建新的逻辑卷。**lvcreate** 命令中至少包含用于设置 LV 名称的 **-n** 选项、用于设置 LV 大小（以字节为单位）的 **-L** 选项或用于设置 LV 大小（以区块数为单位）的 **-l** 选项，以及托管此逻辑卷的卷组的名称。

```
[root@host ~]# lvcreate -n lv01 -L 700M vg01
```

这会在 VG **vg01** 中创建一个名为 **lv01**、大小为 700 MiB 的 LV。针对所请求的大小，如果卷组没有足够数量的可用物理区块，此命令将失败。另外请注意，如果大小无法完全匹配，则将四舍五入为物理区块大小的倍数。

您可以使用 **-L** 选项来指定大小，它预期大小单位为字节、兆字节（二进制兆字节，1048576 字节）、千兆字节（二进制千兆字节）等等。或者，您可以使用 **-l** 选项，它预期大小指定为若干个物理区块。

以下列表提供了一些创建 LV 的示例：

- **lvcreate -L 128M**：将逻辑卷的大小确定为正好 128 MiB。
- **lvcreate -l 128**：将逻辑卷的大小确定为正好 128 个区块。字节数取决于基础物理卷上物理区块块的大小。



重要

不同的工具将使用传统名称 `/dev/vgname/lvname` 或内核设备映射程序名 `/dev/mapper/vgname-lvname`, 显示逻辑卷名。

添加文件系统。

使用 `mkfs` 在新逻辑卷上创建 **XFS** 文件系统。或者, 根据您首选的文件系统创建文件系统, 例如 **ext4**。

```
[root@host ~]# mkfs -t xfs /dev/vg01/lv01
```

要使文件系统在重新启动后依然可用, 请执行以下步骤:

- 使用 `mkdir` 创建挂载点。

```
[root@host ~]# mkdir /mnt/data
```

- 向 `/etc/fstab` 文件中添加以下条目:

```
/dev/vg01/lv01 /mnt/data xfs defaults 1 2
```



注意

按名称挂载逻辑卷等同于按 UUID 进行挂载, 因为即使您最初按名称将它们添加到卷组中, LVM 也会根据 UUID 查找其物理卷。

- 运行 `mount /mnt/data`, 挂载刚刚在 `/etc/fstab` 中添加的文件系统。

```
[root@host ~]# mount /mnt/data
```

删除逻辑卷

要删除所有逻辑卷组件, 请执行以下步骤:

准备文件系统。

将必须保留的所有数据移动到另一个文件系统。使用 `umount` 命令卸载文件系统, 然后删除与该文件系统关联的所有 `/etc/fstab` 条目。

```
[root@host ~]# umount /mnt/data
```



警告

删除逻辑卷将会破坏该逻辑卷上存储的所有数据。删除逻辑卷之前, 请备份或移动数据。

删除逻辑卷。

使用 `lvremove DEVICE_NAME` 删除不再需要的逻辑卷。

```
[root@host ~]# lvremove /dev/vg01/lv01
```

运行此命令之前，卸载 LV 文件系统。在删除 LV 之前，该命令会提示进行确认。

LV 的物理区块会被释放，并可用于分配给卷组中的现有 LV 或新 LV。

删除卷组。

使用 **vgremove VG_NAME** 删除不再需要的卷组。

```
[root@host ~]# vgremove vg01
```

VG 的物理卷会被释放，并可用于分配给系统中的现有 VG 或新 VG。

删除物理卷。

使用 **pvremove** 删除不再需要的物理卷。使用空格分隔的 PV 设备列表同时删除多个 PV。此命令将从分区（或磁盘）中删除 PV 元数据。分区现已空闲，可重新分配或重新格式化。

```
[root@host ~]# pvremove /dev/vdb2 /dev/vdb1
```

查看 LVM 状态信息

物理卷

使用 **pvdisplay** 显示有关物理卷的信息。要列出有关所有物理卷的信息，请使用不带参数的命令。要列出有关特定物理卷的信息，请将相应的设备名称传给该命令。

```
[root@host ~]# pvdisplay /dev/vdb1
--- Physical volume ---
PV Name           /dev/vdb1          ①
VG Name           vg01              ②
PV Size           768.00 MiB / not usable 4.00 MiB ③
Allocatable       yes               ④
PE Size           4.00 MiB
Total PE          191
Free PE           16                ⑤
Allocated PE      175
PV UUID           JWzDpn-LG3e-n2oi-9EtD-VT2H-PMem-1ZXwP1
```

- ① **PV Name** 映射到设备名称。
- ② **VG Name** 显示将 PV 分配到的卷组。
- ③ **PV Size** 显示 PV 的物理大小，包括任何不可用的空间。
- ④ **PE Size** 是物理区块大小，它是逻辑卷中可分配的最小大小。
- ⑤ **Free PE** 显示有多少 PE 单位可用于分配给新逻辑卷。

它也是计算以 PE 单位报告的任何值（如 Free PE）的大小时的倍数；例如： $26 \times 4 \text{ MiB}$ (PE Size) 相当于 104 MiB 可用空间。逻辑卷大小将取整为 PE 单位的倍数。

LVM 会自动设置 PE 大小，但也可以指定该大小。

- ⑤ **Free PE** 显示有多少 PE 单位可用于分配给新逻辑卷。

卷组

使用 **vgdisplay** 显示有关卷组的信息。要列出有关所有卷组的信息，请使用不带参数的命令。要列出有关特定卷组的信息，请将相应的 VG 名称传给该命令。

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name          vg01      ①
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 2
VG Access        read/write
VG Status        resizable
MAX LV
Cur LV          1
Open LV          1
Max PV
Cur PV          2
Act PV          2
VG Size          1016.00 MiB    ②
PE Size          4.00 MiB
Total PE         254       ③
Alloc PE / Size  175 / 700.00 MiB
Free  PE / Size  79 / 316.00 MiB ④
VG UUID          3snNw3-CF71-CcYG-Llk1-p6EY-rHEv-xfUSez
```

- ① **VG Name** 是卷组的名称。
- ② **VG Size** 是存储池可用于逻辑卷分配的总大小。
- ③ **Total PE** 是以 PE 单位表示的总大小。
- ④ **Free PE / Size** 显示 VG 中有多少空闲空间可用于分配给新 LV 或扩展现有 LV。

逻辑卷

使用 **lvdisplay** 显示有关逻辑卷的信息。如果未向命令提供任何参数，则将显示有关所有 LV 的信息；如果提供了 LV 设备名称作为参数，此命令将显示有关该特定设备的信息。

```
[root@host ~]# lvdisplay /dev/vg01/lv01
--- Logical volume ---
LV Path          /dev/vg01/lv01      ①
LV Name          lv01
VG Name          vg01      ②
LV UUID          5IyRea-W8Zw-xLHk-3h2a-IuVN-YaeZ-i3IRrN
LV Write Access  read/write
LV Creation host, time host.lab.example.com, 2019-03-28 17:17:47 -0400
LV Status        available
# open          1
LV Size          700 MiB      ③
Current LE       175       ④
Segments         1
Allocation       inherit
```

```
Read ahead sectors      auto
- current set to        256
Block device            252:0
```

- ① **LV Path** 显示逻辑卷的设备名称。

某些工具可能会将设备名报告为 `/dev/mapper/vgname-lvname`；两个名称都表示同一 LV。

- ② **VG Name** 显示从其分配 LV 的卷组。
③ **LV Size** 显示 LV 的总大小。使用文件系统工具确定可用空间和数据存储的已用空间。
④ **Current LE** 显示此 LV 使用的逻辑区块数。LE 通常映射到 VG 中的物理区块，并因此映射到物理卷。



参考文献

`lvm(8)`、`pvcreate(8)`、`vgcreate(8)`、`lvcreate(8)`、`pvremove(8)`、`vgremove(8)`、
`lvremove(8)`、`pvdisk(8)`、`vgdisplay(8)`、`lvdisplay(8)`、`fdisk(8)`、
`gdisk(8)`、`parted(8)`、`partprobe(8)` 和 `mkfs(8)` man page

▶ 指导练习

创建逻辑卷

在此实验中，您将创建物理卷、卷组、逻辑卷和 XFS 文件系统。此外，您还将持久挂载逻辑卷文件系统。

成果

您应能够：

- 借助 LVM 工具创建物理卷、卷组和逻辑卷。
- 在逻辑卷上创建新文件系统并进行持久挂载。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

在 workstation 上，运行 **lab lvm-creating start** 命令。此命令将运行一个起始脚本，它将确定 servera 计算机是否可从网络访问。此外，它还将验证存储是否可用以及是否安装了相应的软件包。

```
[student@workstation ~]$ lab lvm-creating start
```

- ▶ 1. 使用 **ssh** 命令，以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. 使用 **sudo -i** 命令，切换为 root 用户。student 用户的密码为 student。

```
[student@servera ~]$ sudo -i
[sudo] password for student:
[root@servera ~]#
```

- ▶ 3. 创建物理资源。

- 3.1. 使用 **parted** 创建大小为 256 MiB 的两个分区，并将其类型设置为 Linux LVM。

```
[root@servera ~]# parted -s /dev/vdb mklabel gpt
[root@servera ~]# parted -s /dev/vdb mkpart primary 1MiB 257MiB
[root@servera ~]# parted -s /dev/vdb set 1 lvm on
[root@servera ~]# parted -s /dev/vdb mkpart primary 258MiB 514MiB
[root@servera ~]# parted -s /dev/vdb set 2 lvm on
```

3.2. 使用 **udevadm settle** 让系统注册新分区。

```
[root@servera ~]# udevadm settle
```

► 4. 使用 **pvcreate** 添加两个新分区作为 PV。

```
[root@servera ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
```

► 5. 使用 **vgcreate** 创建由两个 PV 构建的名为 **servera_01_vg** 的新 VG。

```
[root@servera ~]# vgcreate servera_01_vg /dev/vdb1 /dev/vdb2
Volume group "servera_01_vg" successfully created
```

► 6. 使用 **lvcreate** 从 **servera_01_vg** VG 创建一个名为 **servera_01_lv** 的 400 MiB LV。

```
[root@servera ~]# lvcreate -n servera_01_lv -L 400M servera_01_vg
Logical volume "servera_01_lv" created.
```

这将创建一个名为 **/dev/servera_01_vg/servera_01_lv** 但没有文件系统的设备。

► 7. 添加持久文件系统。

7.1. 通过 **mkfs** 命令在 **servera_01_lv** LV 上添加 XFS 文件系统。

```
[root@servera ~]# mkfs -t xfs /dev/servera_01_vg/servera_01_lv
...output omitted...
```

7.2. 在 **/data** 处创建挂载点。

```
[root@servera ~]# mkdir /data
```

7.3. 在 **servera** 上, 将下面这一行添加到 **/etc/fstab** 的末尾:

```
/dev/servera_01_vg/servera_01_lv    /data    xfs    defaults    1 2
```

7.4. 使用 **systemctl daemon-reload** 以新的 **/etc/fstab** 配置来更新 systemd。

```
[root@servera ~]# systemctl daemon-reload
```

7.5. 通过 **mount** 命令验证 **/etc/fstab** 条目并挂载新的 **servera_01_lv** 设备。

```
[root@servera ~]# mount /data
```

► 8. 测试并检查工作。

8.1. 作为最终测试, 将一些文件复制到 **/data** 上, 并验证复制的文件数目。

```
[root@servera ~]# cp -a /etc/*.conf /data
[root@servera ~]# ls /data | wc -l
34
```

在下一个引导式练习中，将验证是否仍具有同样的文件数。

8.2. **parted /dev/vdb print** 将列出 /dev/vdb 上存在的分区。

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size   File system  Name      Flags
 1      1049kB  269MB   268MB          primary   lvm
 2      271MB   539MB   268MB          primary   lvm
```

请注意 **Number** 列，其中包含了值 **1** 和 **2**。它们分别对应于 /dev/vdb**1** 和 /dev/vdb**2**。同时还要注意 **Flags** 列，它指示了分区类型。

8.3. **pvdisplay** 可显示有关每个物理卷的信息。（可选）包含设备名称可仅显示特定 PV 的详细信息。

```
[root@servera ~]# pvdisplay /dev/vdb2
--- Physical volume ---
PV Name           /dev/vdb2
VG Name           servera_01_vg
PV Size           256.00 MiB / not usable 4.00 MiB
Allocatable       yes
PE Size           4.00 MiB
Total PE          63
Free PE           26
Allocated PE      37
PV UUID           2z0Cf3-99YI-w9ny-a1EW-wWhL-S8RJ-M2rfZk
```

此输出显示，PV 已分配给 VG **servera_01_vg**，大小为 256 MiB（尽管 4 MiB 不可用），而物理区块大小 (**PE Size**) 为 4 MiB（最小的可分配 LV 大小）。

存在 63 个 PE，其中 26 个 PE 为空闲，可以在将来分配给 LV，37 个 PE 当前已分配给 LV。这些数据转换为 MiB 值后如下所示：

- 总共 252 MiB (63 个 PE x 4 MiB)；请记住，4 MiB 不可用。
- 空闲 104 MiB (26 个 PE x 4 MiB)
- 已分配 148 MiB (37 个 PE x 4 MiB)

8.4. **vgdisplay vgname** 将显示有关名为 **vgname** 的卷组的信息。

```
[root@servera ~]# vgdisplay servera_01_vg
```

验证以下值：

- **VG Size** 为 **504.00MiB**。
- **Total PE** 为 **126**。
- **Alloc PE / Size** 为 **100 / 400.00MiB**。
- **Free PE / Size** 为 **26 / 104.00MiB**。

8.5. **lvdisplay /dev/vgname/lvname** 将显示有关名为 **lvname** 的逻辑卷的信息。

```
[root@servera ~]# lvdisplay /dev/servera_01_vg/servera_01_lv
```

查看 **LV Path**、**LV Name**、**VG Name**、**LV Status**、**LV Size** 和 **Current LE**（逻辑区块，它们映射到物理区块）。

8.6. **mount** 命令将显示所有已挂载的设备及任何挂载选项。它应当包含 **/dev/servera_01_vg/servera_01_lv**。



注意

许多工具改为报告设备映射器名称 **/dev/mapper/servera_01_vg-servera_01_lv**；它是同一个逻辑卷。

```
[root@servera ~]# mount
```

您应看到（可能在最后一行中）**/dev/mapper/servera_01_vg-servera_01_lv** 挂载于 **/data** 以及相关的挂载信息。

8.7. **df -h** 将显示可读的磁盘可用空间。（可选）包括挂载点可仅显示该文件系统的详细信息。

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  395M   24M  372M   6% /data
```

考虑到文件系统元数据，这些值是我们所预期的结果。

► 9. 从 servera 注销。

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 `workstation` 上，运行 `lab lvm-creating finish` 脚本来结束本练习。此脚本将删除练习过程中在 `servera` 上配置的存储。

```
[student@workstation ~]$ lab lvm-creating finish
```

本引导式练习到此结束。

扩展逻辑卷

培训目标

学完本节后，您应能够：

- 使用 **pvccreate** 和 **vgextend** 扩展卷组 (VG)，并使用 **vgdisplay** 验证结果。
- 使用 **pvmmove** 和 **vgrreduce** 缩减 VG。
- 使用 **lvextend** 扩展逻辑卷 (LV)。
- 通过 **xfs_growfs** 调整 XFS 文件系统的大小。
- 通过 **resize2fs** 调整 ext4 文件系统的大小。

扩展和缩减卷组

您可以通过添加额外的物理卷来为卷组增加更多磁盘空间。这种做法称为扩展卷组。然后，您可以从额外的物理卷中为逻辑卷分配新的物理区块。

未使用的物理卷可以从卷组中删除。这种做法称为缩减卷组。首先，使用 **pvmmove** 命令将数据从一个物理卷上的区块移动到卷组中其他物理卷上的区块。通过这种方式，可以将新磁盘添加到现有卷组，将数据从较旧或较慢的磁盘移动到新磁盘，并将旧磁盘从卷组中删除。可在卷组中的逻辑卷正在使用时执行这些操作。



重要

以下示例使用设备 **vdb** 及其分区来说明 LVM 命令。在实际应用中，请为您自己系统上的磁盘和磁盘分区使用适当的设备。

扩展卷组

要扩展卷组，请执行以下步骤：

准备物理设备并创建物理卷。

就像创建新卷组一样，如果还没有准备好物理卷，则必须创建新分区，并准备好将其用作物理卷。

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1027MiB 1539MiB
[root@host ~]# parted -s /dev/vdb set 3 lvm on
[root@host ~]# pvccreate /dev/vdb3
```

仅当没有空闲的 PV 可以扩展 VG 时，才需要创建 PV。

扩展卷组。

使用 **vgextend** 向卷组中添加新物理卷。使用 VG 名称和 PV 设备名称作为 **vgextend** 的参数。

```
[root@host ~]# vgextend vg01 /dev/vdb3
```

此命令会对 **vg01** VG 进行扩展，扩展幅度为 **/dev/vdb3** PV 的大小。

验证新空间是否可用。

使用 **vgdisplay** 确认额外的物理区块是否可用。检查输出中的 **Free PE / Size**。它不应当为零。

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name          vg01
...output omitted...
Free  PE / Size    178 / 712.00 MiB
...output omitted...
```

缩减卷组

要缩减卷组，请执行以下步骤：

移动物理区块。

使用 **pvmove PV_DEVICE_NAME** 将要删除的物理卷中的所有物理区块都重新放置到卷组中的其他物理卷上。其他物理卷中必须有足够的空闲区块来容纳这些移动内容。仅当 VG 中存在足够的空闲区块，且所有这些区块都来自其他 PV 时，才能执行此操作。

```
[root@host ~]# pvmove /dev/vdb3
```

此命令会将 PE 从 **/dev/vdb3** 移动到同一 VG 中具有空闲 PE 的 PV。



警告

使用 **pvmove** 前，备份卷组中所有逻辑卷上存储的数据。如果操作期间意外断电，可能会导致卷组状态不一致。这可能导致卷组中逻辑卷上的数据丢失。

缩减卷组。

使用 **vgreduce VG_NAME PV_DEVICE_NAME** 从卷组中删除物理卷。

```
[root@host ~]# vgreduce vg01 /dev/vdb3
```

它将从 **vg01** VG 中删除 **/dev/vdb3** PV，并可以添加到其他 VG。或者，也可以使用 **pvremove** 永久停止将设备用作 PV。

扩展逻辑卷和 XFS 文件系统

逻辑卷的一个优势在于能够在不停机的情况下增加其大小。可将卷组中的空闲物理区块添加到逻辑卷以扩展其容量，然后可使用逻辑卷扩展所包含的文件系统。

扩展逻辑卷

要扩展逻辑卷，请执行以下步骤：

验证卷组是否具有可用的空间。

使用 **vgdisplay** 验证是否有足够的物理区块可供使用。

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name          vg01
...output omitted...
Free PE / Size   178 / 712.00 MiB
...output omitted...
```

检查输出中的 **Free PE / Size**。确认卷组中是否有足够的空闲空间可用于 LV 扩展。如果可用空间不足，则相应地扩展卷组。请参见“扩展和缩减卷组”一节。

扩展逻辑卷。

使用 **lvextend** 将逻辑卷扩展为新的大小。

```
[root@host ~]# lvextend -L +300M /dev/vg01/lv01
```

此命令会将逻辑卷 **lv01** 的大小增加 300 MiB。请注意大小前面的加号 (+)，它表示向现有大小增加此值；如无该符号，该值定义 LV 的最终大小。

和 **lvcreate** 一样，存在不同的方法来指定大小：-l 选项预期以物理区块数作为参数。-L 选项则预期以大小（单位为字节、兆字节、千兆字节等等）作为参数。

以下列表提供了一些扩展 LV 的示例。

扩展 LV 的示例

命令	结果
lvextend -l 128	将逻辑卷的大小调整为正好 128 个区块。
lvextend -l +128	向逻辑卷的当前大小添加 128 个区块。
lvextend -L 128M	将逻辑卷的大小调整为正好 128 MiB。
lvextend -L +128M	向逻辑卷的当前大小添加 128 MiB。
lvextend -l +50%FREE	向 LV 添加 VG 中当前可用空间的 50%。

扩展文件系统。

使用 **xfs_growfs mountpoint** 可以扩展文件系统以占用已扩展的 LV。使用 **xfs_growfs** 命令时，必须挂载目标文件系统。在调整文件系统大小时，可以继续使用该文件系统。

```
[root@host ~]# xfs_growfs /mnt/data
```



注意

常见错误是运行 **lvextend**，但忘记运行 **xfs_growfs**。连续运行两个步骤的一种替代方法是，在 **lvextend** 命令中包含 -r 选项。这将使用 **fsadm(8)** 在扩展 LV 后调整文件系统的大小。它可以用于多种不同的文件系统。

· 验证已挂载文件系统的新大小：

df -h /mountpoint.

扩展逻辑卷和 EXT4 文件系统

扩展基于 **ext4** 的逻辑卷的步骤基本与扩展基于 **XFS** 的 LV 相同，只有调整文件系统大小的步骤不同。检查“扩展逻辑卷和 XFS 文件系统”一节。

验证卷组是否具有可用的空间。

使用 **vgdisplay VGNAME** 验证卷组中是否有足够数量的物理区块可供使用。

扩展逻辑卷。

使用 **lvextend -l +extents /dev/vgname/lvname** 对逻辑卷 /dev/vgname/lvname 进行扩展，扩展的幅度为 extents 值。

扩展文件系统。

使用 **resize2fs /dev/vgname/lvname** 可以扩展文件系统以占用新扩展的 LV。运行扩展命令时，可以挂载并使用文件系统。可以包含 **-p** 选项以监控调整大小操作的进度。

```
[root@host ~]# resize2fs /dev/vg01/lv01
```



注意

xfs_growfs 与 **resize2fs** 之间的主要区别是为识别文件系统而传递的参数。**xfs_growfs** 采用挂载点，而 **resize2fs** 采用逻辑卷名称。

扩展逻辑卷和交换空间

格式化为交换空间的逻辑卷也可以进行扩展，但该过程与扩展文件系统（如 **ext4** 或 **XFS**）的过程有所不同。格式化为相应文件系统的逻辑卷可进行动态扩展，无需停机。格式化为交换空间的逻辑卷则必须脱机才能进行扩展。

验证卷组是否具有可用的空间。

使用 **vgdisplay vgname** 验证是否有足够数量的空闲物理区块可供使用。

停用交换空间。

使用 **swapoff -v /dev/vgname/lvname** 可以停用逻辑卷上的交换空间。



警告

您的系统必须有足够的可用内存或交换空间，以便在停用逻辑卷上的交换空间时能接受需要置入的任何内容。

扩展逻辑卷。

lvextend -l +extents /dev/vgname/lvname 可对逻辑卷 /dev/vgname/lvname 进行扩展，扩展的幅度为 extents 值。

将逻辑卷格式化为交换空间。

mkswap /dev/vgname/lvname 可将整个逻辑卷格式化为交换空间。

激活交换空间。

使用 **swapon -va /dev/vgname/lvname** 可以激活逻辑卷上的交换空间。



参考文献

lvm(8)、**pvccreate(8)**、**pvmove(8)**、**vgdisplay(8)**、**vgextend(8)**、**vgreduce(8)**、
vgdisplay(8)、**vgextend(8)**、**vgreduce(8)**、**lvextend(8)**、**fdisk(8)**、
gdisk(8)、**parted(8)**、**partprobe(8)**、**xfs_growfs(8)** 和 **resize2fs(8)**
swapoff(8) **swapon(8)** **mkswap(8)** man page

► 指导练习

扩展逻辑卷

在本实验中，您将扩展上一练习中添加的逻辑卷。

成果

您应能够：

- 扩展卷组以包含更多物理卷。
- 在文件系统仍处于挂载和使用状态时，调整逻辑卷的大小。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

在 workstation 上，运行 **lab lvm-extending start** 命令。此命令将运行一个起始脚本，它将确定主机 servera 是否可从网络访问，并确保上一个引导式练习中的存储可供使用。

```
[student@workstation ~]$ lab lvm-extending start
```

► 1. 使用 ssh 命令，以 student 用户身份登录 servera。

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

► 2. 使用 sudo -i 命令，在 shell 提示符处切换为 root。

```
[student@servera ~]$ sudo -i  
[sudo] password for student:  
[root@servera ~]#
```

► 3. 使用 vgdisplay 确定 VG 是否具有足够的可用空间，可将 LV 扩展至总大小 700 MiB。

```
[root@servera ~]# vgdisplay servera_01_vg  
--- Volume group ---  
VG Name           servera_01_vg  
System ID  
Format            lvm2  
...output omitted...  
VG Size           504.00 MiB  
PE Size           4.00 MiB  
Total PE          126
```

```

Alloc PE / Size      100 / 400.00 MiB
Free  PE / Size     26 / 104.00 MiB
VG UUID              OBBAfU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV

```

现在只有 104 MiB 可用（26 个 PE x 4 MiB 区块），至少还需要 300 MiB 才能让总大小达到 700 MiB。您需要扩展 VG。

为便于稍后进行比较，使用 **df** 记录当前磁盘可用空间：

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv 395M   24M  372M   6% /data
```

► 4. 创建物理资源。

4.1. 使用 **parted** 创建 512 MiB 的额外分区，并将其类型设置为 Linux LVM。

```
[root@servera ~]# parted -s /dev/vdb mkpart primary 515MiB 1027MiB
[root@servera ~]# parted -s /dev/vdb set 3 lvm on
```

4.2. 使用 **udevadm settle** 让系统注册新分区。

```
[root@servera ~]# udevadm settle
```

► 5. 使用 **pvcreate** 将新分区添加为 PV。

```
[root@servera ~]# pvcreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

► 6. 扩展卷组。

6.1. 使用新的 /dev/vdb3 PV，通过 **vgextend** 扩展名为 servera_01_vg 的 VG。

```
[root@servera ~]# vgextend servera_01_vg /dev/vdb3
Volume group "servera_01_vg" successfully extended
```

6.2. 再次使用 **vgdisplay** 检查 servera_01_vg VG 的可用空间。现在应该有大量可用空间。

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name          servera_01_vg
System ID
Format          lvm2
...output omitted...
VG Size          1012.00 MiB
PE Size          4.00 MiB
Total PE         253
Alloc PE / Size  100 / 400.00 MiB
Free  PE / Size  153 / 612.00 MiB
VG UUID          OBBAfU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

现在有 612 MiB 的可用空间（153 个 PE x 4 MiB 区块）。

► 7. 使用 **lvextend** 将现有 LV 扩展到 700 MiB。

```
[root@servera ~]# lvextend -L 700M /dev/servera_01_vg/servera_01_lv
  Size of logical volume servera_01_vg/servera_01_lv changed from 400.00 MiB (100
  extents) to 700.00 MiB (175 extents).
  Logical volume servera_01_vg/servera_01_lv successfully resized.
```



注意

本示例指定了最终 LV 的确切大小，但您可能已指定所需的额外空间量：

- **-L +300M**，采用以 MiB 为单位的大小添加新空间。
- **-l 175**，指定总范围数（175 个 PE x 4 MiB）。
- **-l +75**，添加所需的额外范围数。

► 8. 使用 **xfs_growfs** 将 XFS 文件系统扩展到 LV 上的其余可用空间。

```
[root@servera ~]# xfs_growfs /data
meta-data=/dev/mapper/servera_01_vg-servera_01_lv isize=512      agcount=4,
  agsize=25600 blks
...output omitted...
```

► 9. 使用 **df** 和 **ls | wc** 查看新文件系统大小，并验证之前的现有文件是否仍然存在。

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  695M   26M  670M   4% /data
[root@servera ~]# ls /data | wc -l
34
```

文件仍然存在，文件系统近似于指定的大小。

► 10. 从 **servera** 注销。

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 **workstation** 上，运行 **lab lvm-extending finish** 命令来结束本练习。此脚本将删除练习过程中在 **servera** 上配置的存储。

```
[student@workstation ~]$ lab lvm-extending finish
```

本引导式练习到此结束。

▶ 开放研究实验

管理逻辑卷

任务执行清单

在本实验中，您将调整现有逻辑卷的大小，同时根据需要添加 LVM 资源，然后添加一个新逻辑卷，该逻辑卷上带有持久挂载的 XFS 文件系统。

成果

您应能够：

- 将 `serverb_01_lv` 逻辑卷的大小调整为 768 MiB。
- 创建一个名为 `serverb_02_lv` 的 128 MiB 新逻辑卷，带有 XFS 文件系统，持久挂载于 `/storage/data2`。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 `lab lvm-review start` 命令。此命令将运行一个起始脚本，它将确定 `serverb` 计算机是否可从网络访问。此外，它还在 `serverb` 上准备了存储空间，供练习使用。

```
[student@workstation ~]$ lab lvm-review start
```

在 `serverb` 上，一个挂载于 `/storage/data1` 且名为 `serverb_01_lv` 的逻辑卷出现磁盘空间不足的情况，现在要求您将其扩展为 768 MiB 的大小。您必须确保 `serverb_01_lv` 始终持久挂载于 `/storage/data1`。

此外，您还需要创建一个名为 `serverb_02_lv` 的 128 MiB 新逻辑卷，且挂载于 `/storage/data2`。您接到指示，需要将新逻辑卷格式化为 XFS 文件系统。

这些逻辑卷位于 `serverb_01_vg` 卷组中。但遗憾的是，该卷组中没有足够的空间来扩展现有的逻辑卷和添加新逻辑卷。之前在 `/dev/vdb` 上创建了一个 512 MiB 分区。您接到指示，需要 `/dev/vdb` 上再使用 512 MiB。您必须创建新的分区。

1. 在 `/dev/vdb` 上创建一个 512 MiB 分区，将其初始化为物理卷，然后使用它来扩展 `serverb_01_vg` 卷组。
 2. 将 `serverb_01_lv` 逻辑卷扩展到 768 MiB，包括文件系统。
 3. 在现有卷组中，创建名为 `serverb_02_lv` 且大小为 128 MiB 的一个新逻辑卷。添加 XFS 文件系统，并将其永久挂载于 `/storage/data2`。
 4. 完成后，重新启动 `serverb` 计算机，并从 `workstation` 计算机运行命令 `lab lvm-review grade` 以验证工作。
- 等待 `serverb` 完全启动，然后继续执行评估。

评估

在 workstation 上，运行 **lab lvm-review grade** 脚本来确认是否成功完成本练习。

```
[student@workstation ~]$ lab lvm-review grade
```

完成

在 workstation 上，运行 **lab lvm-review finish** 脚本来完成实验。

```
[student@workstation ~]$ lab lvm-review finish
```

本实验到此结束。

► 解决方案

管理逻辑卷

任务执行清单

在本实验中，您将调整现有逻辑卷的大小，同时根据需要添加 LVM 资源，然后添加一个新逻辑卷，该逻辑卷上带有持久挂载的 XFS 文件系统。

成果

您应能够：

- 将 `serverb_01_lv` 逻辑卷的大小调整为 768 MiB。
- 创建一个名为 `serverb_02_lv` 的 128 MiB 新逻辑卷，带有 XFS 文件系统，持久挂载于 `/storage/data2`。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 `lab lvm-review start` 命令。此命令将运行一个起始脚本，它将确定 `serverb` 计算机是否可从网络访问。此外，它还在 `serverb` 上准备了存储空间，供练习使用。

```
[student@workstation ~]$ lab lvm-review start
```

在 `serverb` 上，一个挂载于 `/storage/data1` 且名为 `serverb_01_lv` 的逻辑卷出现磁盘空间不足的情况，现在要求您将其扩展为 768 MiB 的大小。您必须确保 `serverb_01_lv` 始终持久挂载于 `/storage/data1`。

此外，您还需要创建一个名为 `serverb_02_lv` 的 128 MiB 新逻辑卷，且挂载于 `/storage/data2`。您接到指示，需要将新逻辑卷格式化为 XFS 文件系统。

这些逻辑卷位于 `serverb_01_vg` 卷组中。但遗憾的是，该卷组中没有足够的空间来扩展现有的逻辑卷和添加新逻辑卷。之前在 `/dev/vdb` 上创建了一个 512 MiB 分区。您接到指示，需要 `/dev/vdb` 上再使用 512 MiB。您必须创建新的分区。

1. 在 `/dev/vdb` 上创建一个 512 MiB 分区，将其初始化为物理卷，然后使用它来扩展 `serverb_01_vg` 卷组。

- 1.1. 以 `student` 用户身份登录 `serverb`。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. 使用 `sudo -i` 命令，切换为 `root` 用户。`student` 用户的密码为 `student`。