项目3

企业级存储扩展与数据安全

CHAPTER 3

1. 知识目标

（1）掌握 RAID 10 的技术原理、适用场景及配置逻辑。

（2）理解 LVM 的核心机制及快照功能。

（3）了解自动化备份的常见策略及工具。

2. 能力目标

（1）能根据业务需求规划存储扩展方案。

（2）能完成 LVM 逻辑卷的创建、扩容及快照配置，实现数据灵活管理。

（3）能设计并实施自动化备份脚本，定期验证备份有效性。

视频讲解

3. 德育目标

（1）树立“数据是核心资产”的理念，强化数据保护的责任意识。

（2）培养风险预判能力，在技术实施前评估潜在安全隐患。

3.1 项目描述

在当今数字化时代,行业数据呈现出爆炸式增长的态势。据权威机构统计,过去五年间,企业数据量以平均每年 50%的速度递增,预计未来三年这一增速还将进一步加快。企业在享受数据带来价值的同时,也面临着诸多存储痛点。一方面,存储容量不足的问题日益凸显,许多企业的现有存储设备利用率已超过 80%,接近饱和状态,难以为新业务的开展提供支持;另一方面,数据安全事件频发,给企业带来了巨大的损失。据不完全统计,每年因数据泄漏、勒索攻击等安全事件造成的经济损失高达数亿美元。因此,启动企业级存储扩展与数据安全项目具有重要的战略意义。该项目不仅能够满足企业日益增长的存储需求,保障业务的持续稳定运行,还能提升企业的数据安全防护能力,增强企业的核心竞争力。

某企业核心业务服务器面临存储容量不足问题,同时需提升关键业务数据(如数据库、共享文件)的可靠性。传统物理分区扩展需停机维护,且单盘存储存在单点故障风险。本项目通过 LVM 逻辑卷管理实现动态存储扩展,并采用 RAID 10 技术保障数据安全,满足业务连续性需求。

本项目目标如下所示。

(1) 动态扩展存储。在不中断业务的情况下扩容磁盘容量。

(2) 数据高可用性。通过 RAID 10 实现冗余与故障自动恢复。

(3) 自动化管理。结合 LVM 快照与 cron 定时任务实现定期备份。

3.2 资源环境

本项目依托 Windows 10 个人主机作为底层平台,通过 VMware Workstation Pro 虚拟化技术构建实验与运行环境。整体硬件与软件环境围绕“企业级存储扩展与数据安全”场景的需求进行针对性设计与验证。

3.2.1 硬件环境

本项目运行的硬件环境为:在搭载 Windows 10 操作系统的个人主机上,通过 VMware Workstation Pro 创建虚拟机。模拟配置:CPU 为 2 颗虚拟处理器(每颗 4 核),内存 8GB,存储配置 60GB、20GB、20GB 三块硬盘,操作系统选用 Red Hat Enterprise Linux 9.4。硬件环境如表 3-1 所示。

表 3-1 硬件环境

组 件	最低配置	推荐配置
CPU	x86-64 架构,4 核	4 核及以上
内存	4GB	8GB 及以上
存储	60GB	100GB 及以上(SSD 推荐)
网络	千兆以太网	千兆/万兆以太网

3.2.2　软件环境

企业级存储扩展与数据安全的运行依赖特定版本的操作系统及基础工具链。项目软件环境如表 3-2 所示，包括支持的 Linux 发行版、文件系统及依赖工具。

表 3-2　项目软件环境

项　　目	版 本 要 求
操作系统	Red Hat Enterprise Linux 9.4
文件系统	XFS(支持在线扩容)
依赖工具	LVM2(逻辑卷管理)、mdadm(软件 RAID 管理)、xfs_growfs/resize2fs(文件系统扩容)、cron(自动化任务调度)

3.3　项目分析

项目分析主要从关键技术选型与潜在风险及应对策略两大维度展开，旨在系统性梳理项目落地的技术路径与可控性保障。前者围绕存储扩展与数据安全等场景，筛选匹配业务规模、性能要求等；后者则针对技术落地过程中的不确定性(如硬件瓶颈、架构复杂度、安全风险)，预判潜在挑战并制订分级应对预案，确保项目推进既具技术前瞻性，又能有效规避或化解关键风险。

3.3.1　关键技术选型

本方案围绕存储管理的灵活性、可靠性、可控性和连续性等核心诉求，从动态扩展、冗余保护、配额管控、备份恢复四个维度选型技术，形成弹性、安全、高效的存储管理体系。关键技术选型如表 3-3 所示。

表 3-3　关键技术选型

需　　求	技 术 方 案	优　　势
动态存储扩展	LVM 逻辑卷管理	支持在线扩容/缩容，无须停机
数据冗余与故障恢复	RAID 10(软/硬件)	兼顾性能与可靠性，允许单盘或多盘故障
存储配额管理	XFS 配额(xfs_quota)	精细化控制用户/组存储空间
自动化备份	LVM 快照 ＋ cron 定时任务	最小化备份窗口，降低业务影响

3.3.2　潜在风险与应对

基于技术方案特性及运维实践，识别两类核心风险并制订针对性防控策略，确保存储系统稳定运行。潜在风险及应对措施如表 3-4 所示。

表 3-4 潜在风险与应对措施

风 险 点	风险描述	应对措施	风险等级
风险 1：RAID 重建期间性能下降	RAID 阵列磁盘故障后，重建过程中需同步数据至新盘，导致 I/O 负载激增，可能影响业务响应速度	① 严格在业务低峰期(如凌晨)执行磁盘更换与重建； ② 提前通过监控工具，如 iostat 评估重建耗时，预留缓冲窗口	中
风险 2：LVM 快照空间溢出失效	LVM 快照依赖原始卷空间，若快照保留时间过长或数据变更量大，可能占满原始卷空间导致快照失效	① 限制快照最大容量，建议不超过原始卷的 20%； ② 结合 cron 任务设置快照自动过期删除(如保留最近 3 次快照)	低

3.4 项目实施

项目按照安装软件的依赖工具、扩展存储空间、RAID 10 配置、LVM 快照与自动化备份、项目验证的流程进行实施。实施过程中，涉及的命令 dnf、lsblk、fdisk、mkfs. xfs、mount、mkdir、cat 在项目 2 中已经介绍，这里不再赘述。下面详细介绍该项目中需要使用的其他命令。

1. 系统管理与包管理

(1) sudo。

基本语法：`sudo [选项] 命令`。

命令详解：以超级用户或其他用户身份执行命令，常用于需要管理员权限的操作(如安装软件、修改系统配置)。例如：`sudo dnf install lvm2` 表示以管理员权限安装 lvm2 工具。

常用选项示例如下。

① -u 用户名。以指定用户身份执行命令。例如`sudo -u fin_user1 cat /data/finance/testfile`以财务用户身份查看文件。

② -i。模拟初始登录(加载目标用户的环境变量)。例如`sudo -i -u dev_user1`切换到 dev_user1 的登录环境。

(2) systemctl。

基本语法：`systemctl [命令] [选项] 服务名`。

命令详解：管理系统服务，可执行启动、停止、重启、查看状态、设置开机自启或禁启等操作。

常用选项示例如下。

① enable --now 服务名。启用并立即启动服务。例如`sudo systemctl enable --now sssd`启用并启动 SSSD 服务。

② status 服务名。查看服务状态。例如`sudo systemctl status sshd`检查 SSH 服务是否运行。

③ restart 服务名。重启服务。例如`sudo systemctl restart nginx`重启 Nginx 服务。

(3) crontab。

基本语法：`crontab [选项]`。

命令详解：管理用户的定时任务(cron job)。

常用选项示例如下。

① -e。编辑当前用户的定时任务。例如`sudo crontab -e`编辑 root 用户的定时任务。

② -l。列出当前用户的定时任务。例如`crontab -l`查看当前用户的所有定时任务。

③ -r。删除当前用户的所有定时任务(谨慎使用)。

2. 存储管理(LVM 相关)

(1) pvcreate。

基本语法:`pvcreate [选项] 物理卷设备路径`。

命令详解:初始化物理磁盘或分区为 LVM 物理卷(PV),使其可被 LVM 管理。例如:`sudo pvcreate /dev/nvme0n2` 将/dev/nvme0n2 初始化为物理卷。

(2) pvdisplay。

基本语法:`pvdisplay [选项] [物理卷设备路径]`。

命令详解:显示物理卷的详细信息(如大小、所属卷组、剩余空间等)。不加设备路径时显示所有物理卷信息。

(3) pvs。

基本语法:`pvs [选项] [物理卷设备路径]`。

命令详解:简洁显示物理卷信息(类似 pvdisplay 的精简版),适合快速查看物理卷列表及关键属性。

(4) vgcreate。

基本语法:`vgcreate [选项] 卷组名 物理卷设备路径`。

命令详解:创建一个卷组(VG),并将指定的物理卷加入该卷组。例如:`sudo vgcreate vg_data /dev/nvme0n2` 创建卷组 vg_data 并添加/dev/nvme0n2。

(5) vgdisplay。

基本语法:`vgdisplay [选项] [卷组名]`。

命令详解:显示卷组的详细信息(如总大小、已用空间、包含的物理卷等)。

(6) vgs。

基本语法:`vgs [选项] [卷组名]`。

命令详解:简洁显示卷组信息(类似 vgdisplay 的精简版),适合快速查看卷组列表及容量概况。

(7) lvcreate。

基本语法:`lvcreate [选项] -n 逻辑卷名 -L 大小 卷组名`。

命令详解:在卷组中创建逻辑卷(LV)。

常用选项示例如下。

① -n 逻辑卷名。指定逻辑卷名称。

② -L 大小。指定逻辑卷大小(如 20GB)。

③ -s。创建快照逻辑卷。例如`sudo lvcreate -s -n snap_data -L 5G /dev/vg_data/lv_data`创建 lv_data 的 5GB 快照。

(8) lvdisplay。

基本语法:`lvdisplay [选项] [逻辑卷路径]`。

命令详解：显示逻辑卷的详细信息(如大小、所属卷组、挂载点等)。

(9) lvs。

基本语法：`lvs [选项] [逻辑卷路径]`。

命令详解：简洁显示逻辑卷信息(类似 lvdisplay 的精简版),适合快速查看逻辑卷列表及容量。

(10) vgextend。

基本语法：`vgextend [选项] 卷组名 物理卷设备路径`。

命令详解：向已有卷组中添加新的物理卷,扩展卷组的可用空间。例如：`sudo vgextend vg_data /dev/nvme0n3`将/dev/nvme0n3 加入卷组 vg_data。

(11) lvextend。

基本语法：`lvextend [选项] -L +大小 逻辑卷路径`。

命令详解：扩展逻辑卷的容量(+号表示增量)。例如：`sudo lvextend -L +20G /dev/vg_data/lv_data` 为 lv_data 增加 20GB 空间。

(12) lvremove。

基本语法：`lvremove [选项] 逻辑卷路径`。

命令详解：删除指定的逻辑卷(需先卸载)。例如：`sudo lvremove -f /dev/vg_data/snap_data` 强制删除快照逻辑卷。

3. RAID 管理

(1) mdadm。

基本语法：`mdadm [模式] [选项] [设备]`。

命令详解：管理 Linux 软件 RAID 阵列的工具。

常用模式与选项如下。

① 创建阵列。`--create /dev/mdX --level=RAID 级别 --raid-devices=磁盘数量 磁盘设备路径`。例如`sudo mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sda /dev/sdb /dev/sdc /dev/sdd`创建 RAID 10 阵列。

② 查看详情。`--detail /dev/mdX`。例如`sudo mdadm --detail /dev/md0`查看 RAID 阵列状态。

③ 故障处理。`/dev/mdX --fail 故障磁盘`、`--remove 故障磁盘`、`--add 新磁盘`。例如`sudo mdadm /dev/md0 --fail /dev/sdc`标记磁盘故障。

4. 文件系统扩容

1) xfs_growfs

基本语法：`xfs_growfs [选项] 逻辑卷路径或挂载点`。

命令详解：扩展 XFS 文件系统的大小(需先扩展逻辑卷)。例如：`sudo xfs_growfs /dev/vg_data/lv_data` 将 lv_data 对应的 XFS 文件系统在线扩容。

文件系统与存储状态查看如下所示。

(1) df。

基本语法：`df [选项] [挂载点或文件路径]`。

命令详解：显示文件系统的磁盘空间使用情况(如总容量、已用空间、可用空间)。

常用选项示例如下。

-h。以人类可读格式显示(如 KB、MB、GB)。例如`df -h /mnt/lv_kingbase`查看挂载点容量。

2) blkid

基本语法：`blkid [选项] [设备路径]`。

命令详解：显示块设备的 UUID(通用唯一识别码)、文件系统类型等信息。例如：`sudo blkid /dev/md0` 获取 RAID 阵列/dev/md0 的 UUID，用于/etc/fstab 自动挂载配置。

5. 文件与配置编辑 vim

基本语法：`vim [选项] 文件路径`。

命令详解：文本编辑器，用于编辑配置文件(如/etc/fstab)。例如：`sudo vim /etc/fstab` 编辑 fstab 实现 RAID 阵列自动挂载。

常用选项示例如下。

① +行号。打开文件并跳转到指定行。例如`vim +10 /etc/sssd/sssd.conf`直接编辑第 10 行。

② +/关键词。打开文件并搜索关键词。例如`vim +/ldap_uri /etc/sssd/sssd.conf`定位到 ldap_uri 配置行。

6. 数据备份与同步 rsync

基本语法：`rsync [选项] 源路径 目标路径`。

命令详解：远程/本地文件同步工具，支持增量备份。

常用选项示例如下。

① -a。归档模式，保留文件属性(权限、时间戳等)。

② -v。详细输出，显示同步过程。

③ -z。压缩传输，节省带宽。

例如：`sudo rsync -avz /mnt/backup /backup_server/kingbase_$(date +%F)` 将快照目录同步到备份服务器，并按日期命名备份目录。

7. 挂载与卸载

(1) mount。

基本语法：`mount [选项] 设备 挂载点`。

命令详解：用于将存储设备(如硬盘、ISO 镜像)挂载到指定目录以访问其内容。

常用选项示例如下。

-o 选项。指定挂载选项。例如`-o loop`表示将文件作为循环设备挂载(常用于 ISO 镜像)；`-o rw`表示可读写挂载。

(2) umount。

基本语法：`umount [选项] 设备或挂载点`。

命令详解：卸载已挂载的文件系统。例如：`sudo umount /mnt/backup` 卸载快照挂载点/mnt/backup。

8. 日期与时间格式化 date

基本语法：`date [选项] [+格式]`。

命令详解：Shell 内置命令，用于格式化日期和时间。

常用格式示例如下。

① +%F。输出当前日期(格式：YYYY-MM-DD)。例如`date +%F`输出"2023-10-01"。

② +%T。输出当前时间(格式：HH:MM:SS)。

常用于备份目录命名，如`kingbase_$(date +%F)`生成"kingbase_2023-10-01"格式的目录名。

3.4.1 安装软件的依赖工具

在 VMware Workstation Pro 17.6 中创建虚拟机，安装 Red Hat Enterprise Linux 9.4 操作系统之后，安装本项目需要的依赖工具 LVM2、mdadm、xfs_growfs/resize2f 和 cron：

```
# 安装 LVM、RAID、XFS 文件系统和定时任务工具
sudo dnf install -y lvm2 mdadm xfsprogs cronie
```

3.4.2 存储扩展

在 Linux 系统中，存储扩展是一项围绕灵活增容、高效利用与平滑扩展展开的系统性工程，其核心逻辑是物理存储抽象化、卷管理逻辑化、文件系统弹性化。首先需识别新增物理存储介质，通过 lsblk、fdisk 等工具确认设备路径与分区状态；随后借助 LVM(逻辑卷管理)将离散的物理卷(PV)聚合为可动态调配的卷组(VG)；再从卷组中按需创建或扩展逻辑卷(LV)，突破传统分区固定大小的局限；完成逻辑卷扩容后，需通过 xfs_growfs(XFS 文件系统)或 resize2fs(ext 系列)等工具实现文件系统的在线扩容，确保业务无感知持续运行。对于多磁盘场景，还可结合 mdadm 构建 RAID 阵列(如 RAID 10)，在扩展容量的同时提升数据冗余与读写性能。下面是本项目存储扩展的具体操作步骤。

步骤 1　添加新虚拟磁盘，默认 20GB，添加后系统中多了一块磁盘 3，目前共有 3 块磁盘。“虚拟机设置”窗口如图 3-1 所示。

步骤 2　启动虚拟机，查看新磁盘的设备名称，将新磁盘格式化为 XFS 文件系统。

```
#列出块设备，确认新添加的磁盘设备名称
lsblk
#查看磁盘分区表
sudo fdisk -l
#格式化新设备为 XFS 文件系统
sudo mkfs.xfs /dev/nvme0n3
```

图 3-1　“虚拟机设置”窗口

步骤 3　在磁盘上创建物理卷。

```
#在第 2 块硬盘/dev/nvme0n2 上创建物理卷
sudo pvcreate /dev/nvme0n2
#验证物理卷创建结果
sudo pvdisplay 或者 sudo pvs
```

步骤 4　创建卷组 vg_data，并把创建好的物理卷/dev/nvme0n2 加入卷组。

```
#创建卷组并添加物理卷
sudo vgcreate vg_data /dev/nvme0n2
#验证卷组创建结果
sudo vgdisplay 或者 sudo vgs
```

步骤 5　在卷组 vg_data 中创建逻辑卷并格式化，命名为 lv_data，大小为 20GB。

```
#创建逻辑卷
sudo lvcreate -n lv_data -L 20G vg_data
#验证卷组创建结果
sudo lvdisplay 或者 sudo lvs
```

```
#格式化逻辑卷
sudo mkfs.xfs /dev/vg_data/lv_data
#将逻辑卷/dev/vg_data/lv_data 挂载到系统中进行使用,挂载目录假设为/mnt/lv_kingbase
sudo mkdir /mnt/lv_kingbase
sudo mount /dev/vg_data/lv_data /mnt/lv_kingbase/
```

步骤 6　将新添加的磁盘/dev/nvme0n3 初始化物理卷,并扩展卷组 vg_data。

```
# 初始化新磁盘为物理卷
sudo pvcreate /dev/nvme0n3
#验证物理卷创建结果
sudo pvdisplay 或者 sudo pvs
# 将物理卷加入现有卷组 vg_data
sudo vgextend vg_data /dev/nvme0n3
# 扩展逻辑卷
sudo lvextend -L +20G /dev/vg_data/lv_data
# 扩展 XFS 文件系统(在线扩容)
sudo xfs_growfs /dev/vg_data/lv_data
```

步骤 7　验证扩展结果。

```
# 检查挂载点容量是否增加
df -h /mnt/lv_kingbase
# 确认逻辑卷大小变化
lsblk
```

3.4.3　数据安全

本项目数据安全首先是从硬件层保障数据安全与性能,利用 RAID 10 配置完成。总体部署是基于业务对高可用、高性能及数据冗余的需求,选用至少 4 块同规格磁盘构建 RAID 10 阵列,完成 RAID 10 配置。然后在 Linux 系统中正确识别并挂载该阵列,确保数据库数据存储于该可靠存储之上。详细配置步骤如下。

步骤 1　创建 RAID 10 阵列。

在关机状态下添加 4 块 SCSI 磁盘,磁盘设备名称分别为/dev/sda、/dev/sdb、/dev/sdc、/dev/sdd。新添加的 4 块磁盘设备信息如图 3-2 所示。

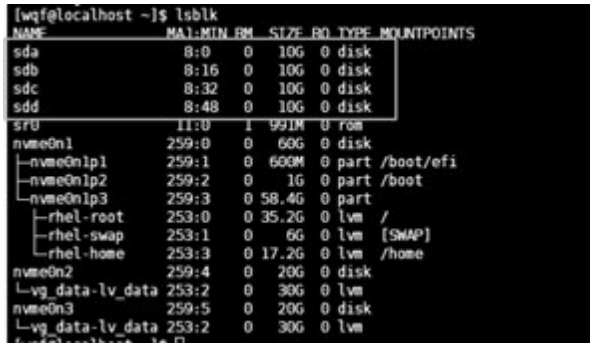

图 3-2　新添加的 4 块磁盘设备信息

```
# 创建磁盘阵列
sudo mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sda /dev/sdb /dev/sdc /dev/sdd
# 为磁盘阵列/dev/md0 格式化为 XFS 文件系统
sudo mkfs.xfs /dev/md0
# 创建磁盘阵列/dev/md0 设备的挂载点
sudo mkdir /mnt/raid10
# 临时挂载磁盘阵列/dev/md0
sudo mount /dev/md0 /mnt/raid10
```

步骤 2　配置 RAID 自动挂载。

```
# 获取 RAID UUID
sudo blkid /dev/md0
[wqf@localhost ~]$ sudo blkid /dev/md0
/dev/md0: UUID="6eb9d490-d608-47e5-a1b4-a113a0fdf647" TYPE="xfs"
# 编辑 /etc/fstab
sudo vim /etc/fstab
# 在/etc/fstab 尾部添加一行,并保存退出:
UUID=6eb9d490-d608-47e5-a1b4-a113a0fdf647 /mnt/raid10 xfs defaults 0 0
# 测试挂载
sudo mount -a
```

步骤 3　监控与故障恢复。

步骤 3-1　查看 RAID 状态。

```
cat /proc/mdstat
sudo mdadm --detail /dev/md0
```

步骤 3-2　模拟磁盘故障并替换(以/dev/sdc 为例)。

```
# 标记磁盘为故障
sudo mdadm /dev/md0 --fail /dev/sdc
# 查看 RAID 的状态,对比操作前后的变化
# sudo mdadm --detail /dev/md0
# 移除故障盘
sudo mdadm /dev/md0 --remove /dev/sdc
# 添加新磁盘(需提前添加好用磁盘/dev/sde)
sudo mdadm /dev/md0 --add /dev/sde
# 查看 RAID 的状态,对比操作前后的变化
# sudo mdadm --detail /dev/md0
```

3.4.4　数据安全增强

在 Linux 系统中,为人大金仓数据库(KingbaseES)进一步增强数据安全,可结合 LVM 快照与自动化备份技术,构建灵活高效的增量保护与灾备能力。总体方案是数据分区采用 LVM 管理,将数据库数据文件所在的逻辑卷纳入 LVM 架构,从而在需要备份或做一致性

保护时,快速创建 LVM 快照。在此基础上,通过编写定时任务(cron)与 LVM 快照联动,实现自动化备份流程。这样既能利用 LVM 快照减少备份窗口、降低对业务性能的影响,又能通过自动化调度保证备份频率与可靠性。

1. 创建 LVM 快照

```
# 创建快照(占用 5GB)
sudo lvcreate -s -n snap_data -L 5G /dev/vg_data/lv_data
# 挂载快照
sudo mount /dev/vg_data/snap_data /mnt/backup
# 同步到备份服务器
sudo rsync -avz /mnt/backup /backup_server/kingbase_$(date +%F)
# 卸载快照
sudo umount /mnt/backup
# 删除快照
sudo lvremove -f /dev/vg_data/snap_data
```

2. 配置 cron 定时任务

步骤 1　编辑定时任务。

```
sudo crontab -e
```

步骤 2　添加每日凌晨 2 点执行的备份脚本。

```
0 2 * * * /usr/local/bin/backup_mysql.sh
```

3. 备份脚本示例(/usr/local/bin/backup_mysql.sh)。

```
#!/bin/bash
# 创建快照
lvcreate -s -n snap_kingbase_$(date +%F) -L 5G /dev/vg_data/lv_data
# 挂载快照
mount /dev/vg_data/snap_kingbase_$(date +%F) /mnt/backup
# 同步到备份服务器
rsync -avz /mnt/backup /backup_server/kingbase_$(date +%F)
# 卸载快照
umount /mnt/backup
# 删除快照
lvremove -f /dev/vg_data/snap_kingbase_$(date +%F)
```

3.5　项目验证

项目部署完成后,需对关键存储与数据安全措施进行全面验证,重点确认存储扩展能力、RAID 数据安全状态及备份机制运行情况,以确保系统具备预期的高可用性与容灾能

力。具体可通过以下操作命令开展验证工作。

1. 存储扩展验证

```
df -h /mnt/lv_kingbase
lsblk
```

2. RAID 数据安全验证

```
# 观察重建进度
cat /proc/mdstat
```

3. 备份验证

```
#检查备份服务器是否生成每日快照
ls -l /backup_server/
```

3.6　项目小结

本项目围绕在 Red Hat Enterprise Linux 9.4 虚拟机环境中实现企业级存储扩展与数据安全项目，通过系统化的实施步骤完成了从环境准备到功能验证的全流程。项目初期明确了硬件需支持动态磁盘扩展、软件需预装 LVM/RAID 工具、网络需保障存储管理通信等资源环境要求，并基于虚拟化平台 VMware 配置了初始虚拟机及新增磁盘资源。实施阶段重点围绕三大核心目标推进：①通过 LVM 动态扩容技术，在不中断业务的前提下将存储容量从初始值扩展至目标值；②采用 RAID 10 技术构建高可用存储阵列，利用 4 块磁盘创建冗余阵列，格式化为 XFS 文件系统后配置自动挂载，同时制定磁盘故障模拟与替换流程，确保数据在硬件故障时的持续可用性；③通过 LVM 快照与定时任务结合的方式强化数据安全，每日定时创建快照、同步至备份服务器，并清理旧快照，形成完整的备份恢复链条，同时配置 cron 定时任务实现自动化执行。项目最终通过多维度验证：存储扩展后容量符合预期、RAID 阵列在模拟磁盘故障时可自动重建、备份任务按计划生成快照且数据可恢复，为企业级虚拟化环境下的存储扩展与数据安全提供了可复用的标准化实施方案。

习题 3