



Lustre 与 NetApp E-Series 存储

NetApp artificial intelligence solutions

NetApp
August 31, 2026

目录

Lustre 与 NetApp E-Series 存储	1
基于 NetApp E 系列存储的 Lustre - 简介	1
解决方案概述	1
Lustre 与 NetApp E 系列存储 - 解决方案架构	1
构建块设计	2
扩展建议	3
HA 架构	4
网络架构	5
Lustre 客户端	5
基于 NetApp E 系列存储的 Lustre - 硬件组件	6
服务器要求	6
EF80 标准构建块	8
存储池选择：TLC 与 QLC	11
网络要求	11
使用 NetApp E 系列存储的 Lustre - 软件组件	11
构建块软件	12
Ansible 自动化	12
Lustre 客户端软件	13
基于 NetApp E 系列存储的 Lustre - 规模调整指南	13
容量估算	13
缩放	14
性能	15
部署解决方案	15
使用 NetApp E-Series 存储部署 Lustre	15
机架和布线 Lustre 硬件	16
准备 Lustre 部署环境	17
自定义 Lustre Ansible 清单	19
部署 Lustre HA 集群和客户端	21
验证和扩展 Lustre 部署	24
使用 NetApp E-Series 存储的 Lustre - 相关资源	25
NetApp 资源	25
Lustre 社区	25
相关 NetApp AI 基础设施解决方案	25

Lustre 与 NetApp E-Series 存储

基于 NetApp E 系列存储的 Lustre - 简介

Lustre with NetApp E-Series Storage 是针对 AI 和 HPC 工作负载的经过验证的并行文件系统解决方案，由 HA 服务器对和 E-Series 全闪存阵列的可重复构建块构建而成。

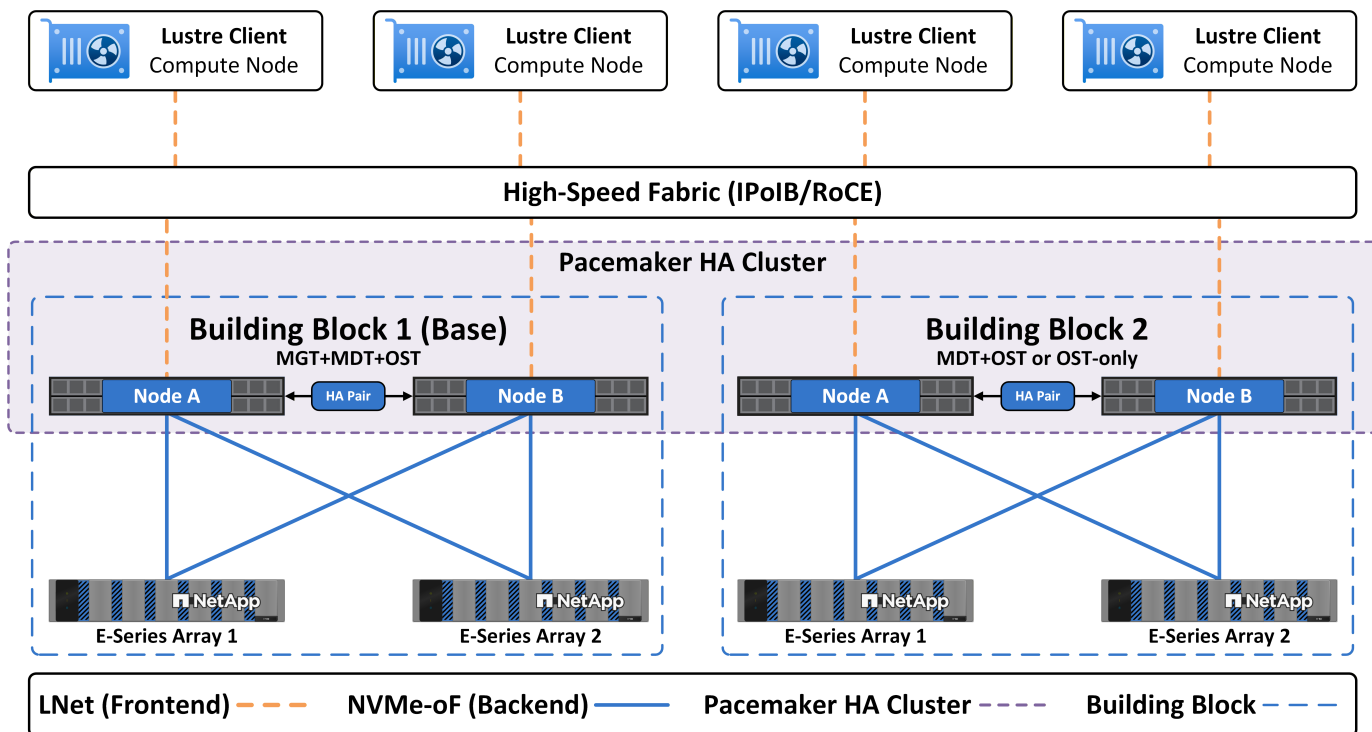
解决方案概述

Lustre 与 NetApp E 系列存储将开源 Lustre 并行文件系统与 NetApp E 系列块存储相结合，为数据密集型工作负载提供可扩展的高性能基础。每个构建块将两个配置为高可用性 (HA) 的 OSS/MDS 服务器节点与两个 E 系列全闪存阵列配对。后端 NVMe-oF 连接将块 I/O 传输到阵列，独立的 LNet 网络连接客户端和 OSS/MDS 服务器节点以进行并行文件系统访问。

基本构建块托管管理服务器 (MGS)、初始元数据目标 (MDTs) 和对象存储目标 (OSTs)。额外的构建块向基本 MGS 注册，以随着工作负载需求的增长扩展元数据容量、数据容量和聚合吞吐量。

下图显示了通过 LNet 连接的多个构建基块和 Lustre 客户端的部署示例。

Lustre with NetApp E-Series 存储解决方案概述



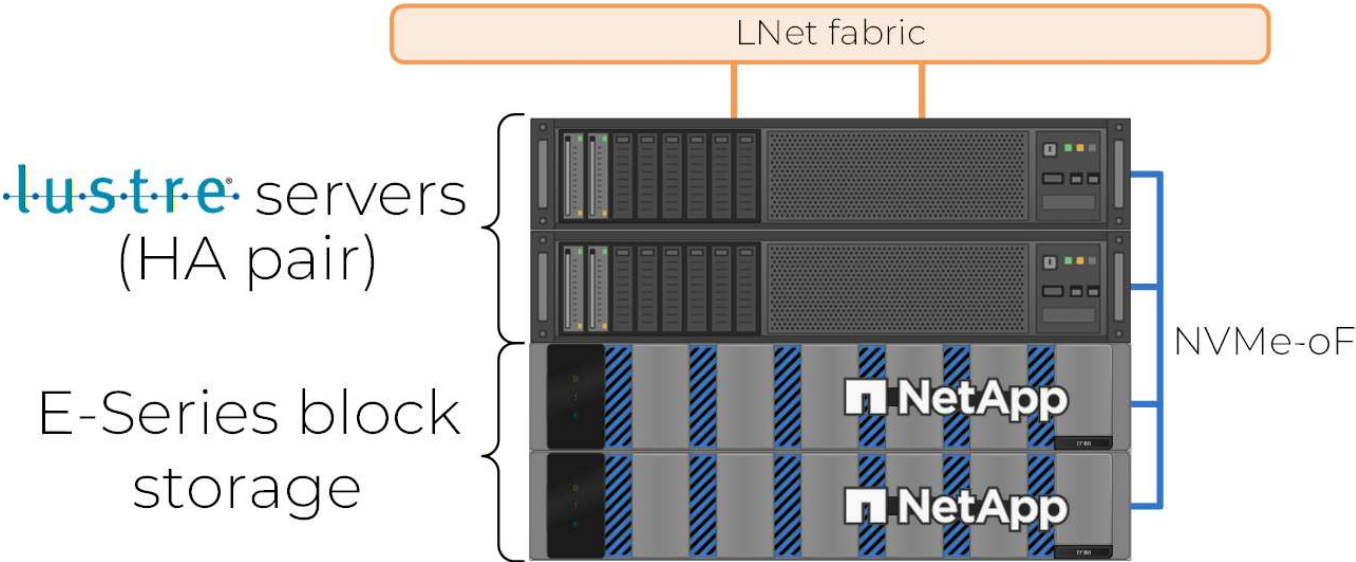
Lustre 与 NetApp E 系列存储 - 解决方案架构

了解 Lustre with NetApp E-Series Storage 如何使用构建块、高可用性、网络拓扑和客户端，以便您可以规划容量、性能和故障转移。

构建块设计

构建模块是 NetApp E 系列 Lustre 解决方案的基本可扩展单元。每个构建块由两个服务器节点组成，配置为提供 Lustre 对象存储服务器 (OSS) 和元数据服务器 (MDS) 功能的高可用性 (HA) 对，两个 NetApp E 系列全闪存阵列，服务器和阵列之间的专用后端 NVMe over Fabrics (NVMe-oF) 连接，以及用于客户端和节点间通信的专用前端 Lustre 网络 (LNet) 连接。Pacemaker 和 Corosync HA 集群可以包含一个或多个构建块中的服务器对。NetApp `ansible-lustre` 集合使用 Ansible 自动化来部署和配置 Lustre 服务。

Lustre 构建块



构建块可根据文件系统的增长需求进行扩展。每个文件系统都需要一个基本构建块。基本构建块在管理目标 (MGT) 上托管管理服务器 (MGS)，并提供初始元数据目标 (MDT) 和对象存储目标 (OST) 容量。其他构建块可扩展文件系统，并将其 MDT 和 OST 目标注册到基本构建块的 MGS 中。模块化方法允许根据工作负载需求独立扩展容量和性能。

NetApp 建议在大多数部署中使用以下构建块配置文件。列出的 MGS、MDT 和 OST 计数是推荐的默认值，经过优化以最大化 E-Series 存储阵列的性能和容量。调整 Ansible 清单中的目标计数、卷大小和 E-Series 驱动器分配，以满足工作负载需求。例如，需要更大元数据存储容量的部署可以在同一构建块硬件中配置更多 MDT 和更少 OST。

下表总结了构建基块类型和建议的目标计数。

类型	MGS	MDT	OSTs	用途
基础	1	8	32	文件系统中的第一个构建块。托管 MGS 和初始 MDT 和 OST 容量。
MDT+OST	0	8	32	添加元数据和数据容量。根据基本构建基块 MGS 进行注册。
仅限 OST	0	0	32	仅添加数据容量。根据基础构建块 MGS 进行注册。

- 驱动器类型和池布局： E-Series 支持传统的 RAID 卷组和动态磁盘池 (DDP)。对于 TLC NVMe 驱动器，将

RAID 6 卷组用于 OST 存储，将 RAID 1 卷组用于 MGS 和 MDT 存储。对于 QLC NVMe 驱动器，请将 DDP 用于共享驱动器池。

- 目标分布：每个构建块中的 Lustre 目标在两个 OSS/MDS 服务器节点和两个 E-Series 阵列之间进行平衡。该布局跨服务器 CPU 和阵列控制器分布 I/O，以提高 NVMe-oF 路径性能并保持冗余。请参阅 ["目标分布和 NVMe-oF 连接"](#) 中的 ["硬件组件"](#)。

支持的操作系统、Lustre 版本、阵列固件和相关构建块组件列于 ["NetApp 互操作性矩阵工具 \(IMT\)"](#) 的 **E-Series Lustre** 下。有关详细的卷计数、驱动器布局 and 大小调整指南，请参阅 ["硬件组件"](#)。有关软件组件，请参阅 ["软件组件"](#)。

扩展建议

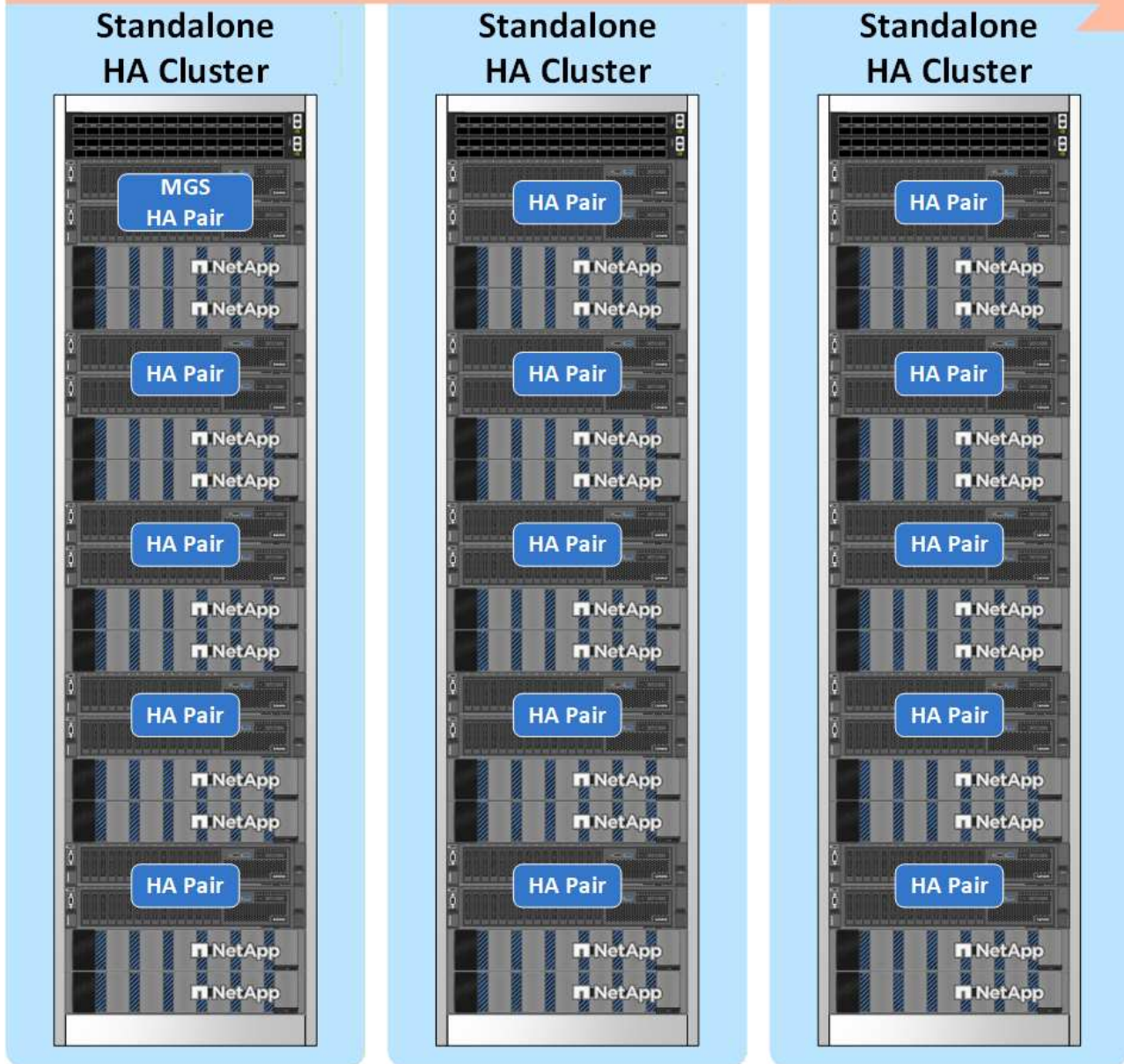
当元数据容量、数据容量或两者都变得有限时，Lustre 文件系统通过添加构建基块进行扩展。根据文件计数和元数据 IOPS 扩展 MDT；根据容量和聚合吞吐量需求扩展 OST。

1. 每个文件系统一个基本构建块：仅部署一个基本构建块；它托管 MGS 以及初始 MDT 和 OST 目标。
2. 其他构建基块配置文件：当现有 MDT 容量足够且数据容量或吞吐量必须增加时，添加仅限 OST 的构建基块。当元数据性能或命名空间容量成为瓶颈时，添加 MDT+OST 构建基块。
3. *HA 集群限制：*将每个 Pacemaker 和 Corosync HA 集群限制为五个构建块（十个 Lustre 服务器节点）。将较大的部署均匀拆分为多个 HA 集群，以避免大型配置中的资源限制。

下图显示了堆叠在 42U 机架中的最多五个构建模块（每个机架一个 Pacemaker HA 集群）。多个机架可以参与单个 Lustre 文件系统；只有基本构建模块托管 MGS。

Lustre 并行文件系统机架扩展

Lustre Parallel Filesystem



有关尺寸示例，请参见 ["尺寸指南"](#)。

HA 架构

Lustre with NetApp E-Series Storage 使用与 NetApp E-Series 存储集成的共享磁盘高可用性 (HA) 架构。Pacemaker 管理 HA 资源，Corosync 提供集群成员资格和消息传递。它们共同管理每个构建块中两个 OSS/MDS 服务器节点之间的 Lustre 目标故障转移。多路径 NVMe-oF 将两个 OSS/MDS 服务器节点连接到相同的 E-Series 卷，因此任一节点都可以在需要时接管目标服务。

每个 MGS、MDT 和 OST 都配置为 HA 资源，其依赖关系位于 Pacemaker 资源组中。Pacemaker 确保资源以正确的顺序启动和停止，保持在同一节点上并置，一次仅在一个节点上运行。从两个节点并发访问同一目标可能会损坏底层文件系统。

- **目标故障转移：**两个 OSS/MDS 服务器节点都是集群中的对等节点，但每个 Lustre 目标一次仅在一个节点上处于活动状态。Pacemaker 监控资源监视每个目标及其依赖项的运行状况，并在目标在其当前节点上不可用时触发故障转移。发生故障时，Pacemaker 会按正确顺序在伙伴节点上重新启动受影响的资源组，一旦服务恢复，客户端将透明地重新连接到目标的新位置。由于每个目标仅在单个节点上激活，因此文件系统永远不会面临来自两个节点的并发写入风险。
- **共享存储访问：**多路径 NVMe-oF 路径将每个 OSS/MDS 服务器节点连接到构建块中的所有 E 系列控制器，因此每个目标卷都可以从任一节点访问。如果服务器节点发生故障，合作伙伴节点已具有指向相同卷的活动路径，无需重新配置存储连接即可接管目标的所有权。如果阵列控制器或路径发生故障，多路径会将 I/O 重定向到存活的控制器的。这种双重冗余使 Lustre 目标能够在 OSS/MDS 服务器节点或阵列控制器之间进行故障切换，而不会丢失对后备卷的访问权限。
- **STONITH 隔离：**发生故障时，Pacemaker 有时无法与故障节点通信以确认其目标已停止。在其他位置重新启动这些目标之前，Pacemaker 会隔离故障节点（理想情况下通过断电），以确保其处于关闭状态。隔离可防止脑裂情况的发生——即两个节点同时访问同一目标并损坏底层文件系统——从而在故障转移期间保持数据完整性。NetApp 建议将 `fence_redfish` 用于配备支持 Redfish 的基板管理控制器（BMC）的服务器。其他隔离代理（例如 `fence_apc`）也受支持。

有关集群管理和隔离配置，请参见 ["HA 用户指南"](#) 中的 ["ansible-lustre 集合"](#)。

网络架构

该解决方案对存储后端流量和 LNet 前端流量使用单独的网络路径。

- **后端 (NVMe-oF)：**OSS/MDS 服务器节点使用 NVMe/InfiniBand 或 NVMe/RoCE 通过 NVMe-oF 连接到 E-Series 阵列。NVMe-oF 路径在 Lustre 目标服务和 E-Series 卷之间传输块 I/O。有关 EF80 六 HCA 后端布线的信息，请参见 ["机架和线缆 Lustre 硬件"](#)。有关跨节点和阵列的首选目标放置，请参见 ["硬件组件"](#) 中的 ["目标分布"](#)。
- **前端 (LNet)：**Lustre 客户端和 OSS/MDS 服务器节点使用 NVIDIA OFED 堆栈通过 LNet 以 `@o2ib` 网络类型进行通信。InfiniBand 和 RoCE 都是经过验证的前端传输协议。`ansible-lustre` 集合为多轨 LNet 配置所有前端接口，通过聚合多个端口来增加带宽，并为客户端及节点间流量提供路径冗余。
- **管理和集群：**带外管理网络提供服务器管理、BMC 访问和阵列管理。STONITH 隔离代理（例如 `fence_redfish`）使用此网络访问服务器 BMC 并从故障节点切断电源。Corosync 还可以在此网络上运行集群通信，作为前端结构旁边的附加环。使用多个环配置 Corosync 可增加集群消息传递的冗余，并降低单个网络故障中断仲裁的风险。

Lustre 客户端

Lustre 客户端加载客户端内核模块，建立到 OSS/MDS 服务器节点的 LNet 连接，并挂载文件系统，以向应用程序提供单一、一致、符合 POSIX 的命名空间。I/O 跨活动 OST 并行分布。客户端节点位于构建块的外部，并通过前端 LNet 结构使用文件系统。此解决方案的 IMT 中未列出客户端操作系统；请使用 ["Lustre 支持表"](#) 查找已部署的 Lustre 版本所支持的已测试客户端发行版和内核版本（例如，Red Hat Enterprise Linux 9、SUSE Linux Enterprise Server 15 和 Ubuntu 24.04）。

客户端节点需要连接到与 OSS/MDS 服务器节点相同的 LNet 结构和网络类型。如果需要，LNet 路由器可以桥接单独的 LNet 子网或结构。

NetApp 为 Rocky Linux 9.8 和 Red Hat Enterprise Linux 9.8 提供预构建的 Lustre 服务器 RPM。NetApp 不提供预构建的客户端软件包。从 ["netapp-lustre 存储库"](#) 中的 Lustre 源代码为客户端操作系统和内核构建客户端软件包。

安装客户端软件包后，["ansible-lustre 集合"](#) 中的可选 `lustre_client` 角色会配置客户端网络接口和 LNet，在

选中时启用 multi-rail LNet，验证连接性，并管理持久的 systemd 挂载单元。该角色不会构建 Lustre。仅当 `lustre_client_packages` 已填充时，它才会安装客户端软件包。如果需要，可以手动配置客户端。有关分步步骤，请参见 [部署解决方案](#) 和 [lustre_client角色文档](#)。

基于 NetApp E 系列存储的 Lustre - 硬件组件

在使用 NetApp E 系列存储调整 and 订购 Lustre 时，请参考服务器、NetApp E 系列阵列、存储布局 and 网络的硬件要求。有关软件组件，请参见[软件组件](#)。

服务器要求

该解决方案使用自带服务器 (BYOS) 模型。每个构建模块需要两个符合或超过以下规范的 OSS/MDS 服务器节点。

节点规格

下表列出了每个 OSS/MDS 节点的最低服务器要求。

组件	需求
数量	每个构建基块 2 个节点
CPU	AMD EPYC 或 Intel Xeon，32 核或更高
内存	256 GB DDR5（最低）
网络 HCA	6 个双端口 200Gb HCA（请参见 HCA 连接 ）
PCIe插槽	2× PCIe Gen5 x16 和 4× PCIe Gen5 x8（请参阅 PCIe 插槽要求 ）
启动驱动器	2× 个驱动器，采用 RAID 1（建议使用软件或硬件 RAID）

NetApp 使用 Lenovo ThinkSystem SR665 V3 服务器验证了该解决方案。当其他供应商的等效服务器满足这些要求时，即可获得支持。

HCA 连接

下表列出了每个 OSS/MDS 服务器节点的 HCA、LNet 前端端口和 NVMe-oF 路径要求。

每个节点的 HCA	每个 Lustre 节点的 LNet 端口	每个 Lustre 节点的 NVMe-oF 路径	示例 HCA
6（12 个端口）	4	共 8 个（每个数组 4 个）	MCX755106AS-HEAT（双端口 200Gb PCIe gen5 卡）

NetApp 建议每个节点配置六个 HCA，以最大化 EF80 存储阵列的带宽。

PCIe插槽要求

EF80 构建模块中的每个 OSS/MDS 服务器节点容纳六个双端口 HCA：两个用于 LNet，四个用于 NVMe-oF。

插槽宽度决定了每个 HCA 可用的带宽，因此请确认每个插槽和转接卡的电气宽度，而不仅仅是卡本身的宽度。安装在 Gen5 x8 插槽中的 Gen5 x16 卡以 x8 速度运行。

- **LNNet HCAs:** 安装在 PCIe Gen5 x16 插槽中，以达到完整的前端吞吐量。
- **NVMe-oF HCAs:** 安装在 PCIe Gen5 x8 或 PCIe Gen5 x16 插槽中。
- ***NUMA balance:** *在两个 NUMA 区域之间均匀划分 HCA，以便每个区域托管两个 LNet 接口和四个 NVMe-oF 接口。

下表列出了 EF80 构建块中每个 OSS/MDS 服务器节点的最小插槽数。

流量类型	每个节点的 HCA	最小插槽宽度	每个 NUMA 区域的插槽数
LNNet	2	PCIe Gen5 x16	1
NVMe-oF	4	PCIe Gen5 x8	2

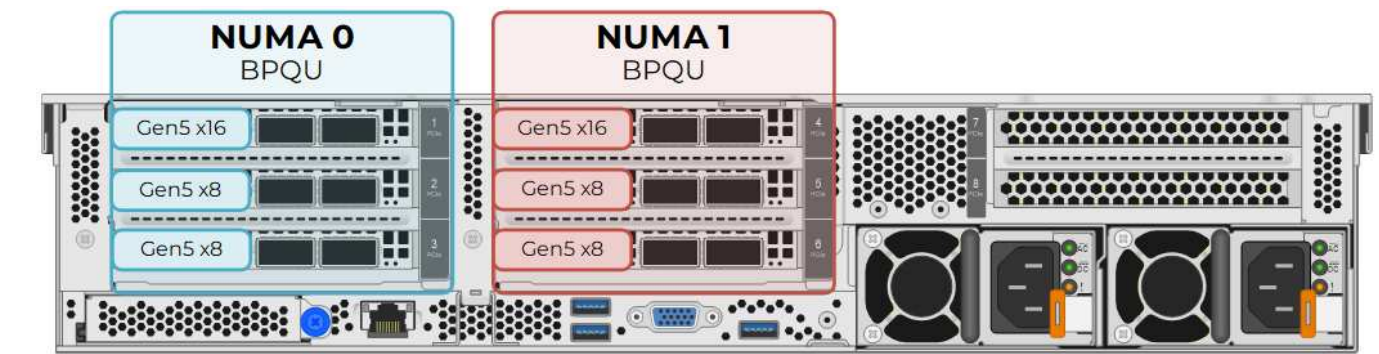
您也可以选择 在双端口 HCA 上拆分端口，以便一个端口传输 LNet 流量，另一个端口传输 NVMe-oF 流量。将所有四个 HCA 安装在 PCIe Gen5 x16 插槽中，以便每个 LNet 端口达到完整吞吐量，然后在 Gen5 x8 或 Gen5 x16 插槽中为其余 NVMe-oF 端口额外添加两个 HCA。

▼ 联想 ThinkSystem SR665 V3 的转接卡配置示例

以下两种提升管组合均满足 EF80 构建模块的插槽要求。

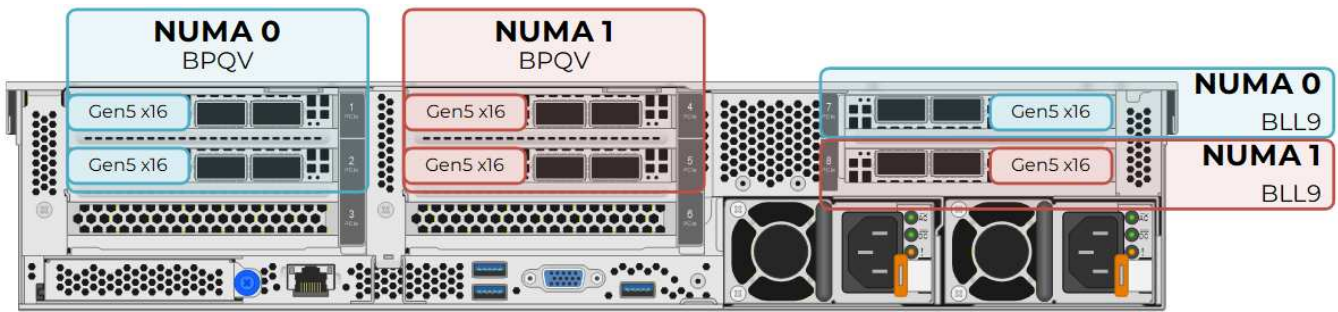
最小插槽宽度

在转接卡位置 1 和 2 中安装 BPQU 转接卡。每个 BPQU 转接卡提供一个 PCIe Gen5 x16 插槽和两个 PCIe Gen5 x8 插槽。这两个转接卡合计为 LNet 提供两个 Gen5 x16 插槽，为 NVMe-oF 提供四个 Gen5 x8 插槽，在 NUMA 区域之间均匀分配。



所有第5代 x16 插槽

在提升板位置 1 和 2 安装 BPQV 提升板，在提升板位置 3 安装 BLL9 提升板。每个 BPQV 提升板提供两个 PCIe Gen5 x16 插槽。BLL9 提升板在 NUMA 区域 0 上提供插槽 7，在 NUMA 区域 1 上提供插槽 8，两者均为 PCIe Gen5 x16。此组合提供六个 Gen5 x16 插槽，每个 NUMA 区域三个插槽，并支持在同一 HCA 的端口上分流 LNet 和 NVMe-oF 流量。



EF80 标准构建块

EF80 是此解决方案版本经过验证的标准平台。

阵列规格

下表列出了构建基块（两个阵列）的 EF80 阵列规格。

组件	规格
型号	NetApp EF80
外形规格	2U 基本机箱，24 个内置 NVMe SSD 插槽
控制器	双控制器（A 和 B）
驱动器	每个阵列 24× NVMe SSD
I/O 连接	在经过验证的 Lustre 设计中，每个阵列 8 个 200Gb NVMe/IB 或 NVMe/RoCE 主机端口

当每个控制器中安装三个双端口主机 I/O 模块时，EF80 平台每个阵列最多支持十二个主机端口。经过验证的 Lustre 设计将插槽 1 和 2 中的主机 I/O 模块用于每个阵列八个主机端口。

每个 EF80 阵列还使用专用插槽 4 I/O 模块进行控制器间镜像。将控制器 A 端口 4a 连接到控制器 B 端口 4a，将控制器 A 端口 4b 连接到控制器 B 端口 4b。这些连接提供缓存镜像和 I/O 传输，不用于主机 NVMe-oF 流量。请参阅 ["用缆线连接 EF50 和 EF80 控制器间镜像连接"](#)。

有关所有型号的完整 EF 系列规格，请参见 ["NetApp EF-Series 全闪存阵列数据表"](#)。

驱动器布局（每个阵列24个驱动器）

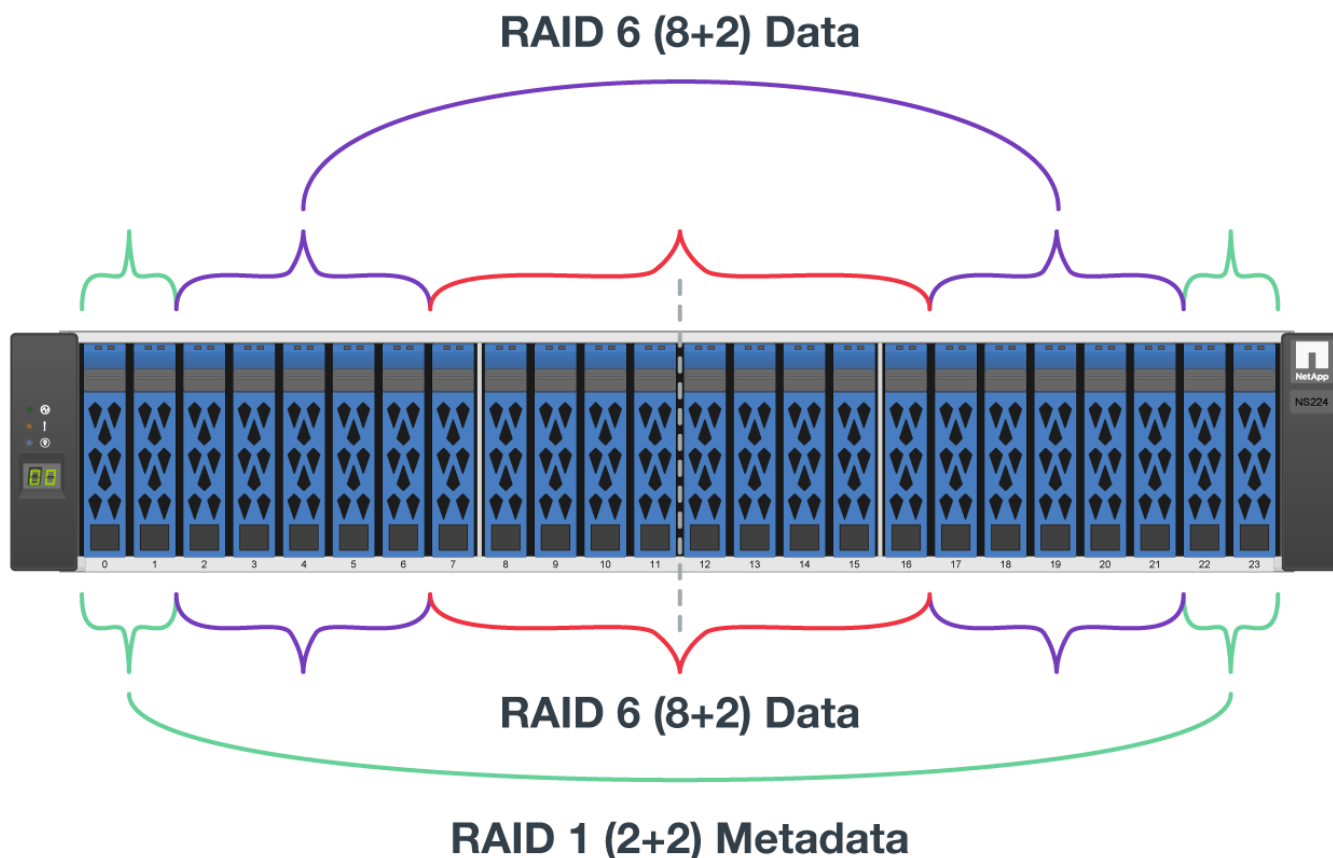
基本构建基块中的每个 EF80 阵列都使用全部 24 个 NVMe 驱动器插槽：

- 4 个驱动器位于 RAID 1 中，用于 MGS/MDT 存储（共享卷组或 DDP 分配）
- RAID 6 中的 10 个驱动器用于 OST 存储（第一个 OST 池）
- RAID 6 中的 10 个驱动器用于 OST 存储（第二个 OST 池）

此布局适用于将 3.84 TB、7.68 TB 或 15.3 TB 驱动器与传统卷组一起使用时。使用 30.7 TB 或 61.4 TB 容量闪存 (QLC) 驱动器时，在所有 24 个驱动器上配置单个动态磁盘池，并在池内为 MGS 和 MDT 卷创建 RAID 1 卷，同时为 OST 使用默认 RAID 6 卷。



目前不建议将 1.92 TB 驱动器用于此解决方案。请使用上面列出的经过验证的驱动器容量之一。



卷和目标计数（基本构建块）

下表列出了基本构建基块的 MGS、MDT 和 OST 计数。

目标类型	每 BB 计数	RAID / 池	备注
MGS	1	RAID 1	仅基础构建基块；数组 1
MDT	8	RAID 1	每个阵列 4 个
OST	32	RAID 6 (TLC) 或 DDP (QLC)	每个阵列 16 个

使用以下卷大小调整准则作为 Ansible 清单的起点。

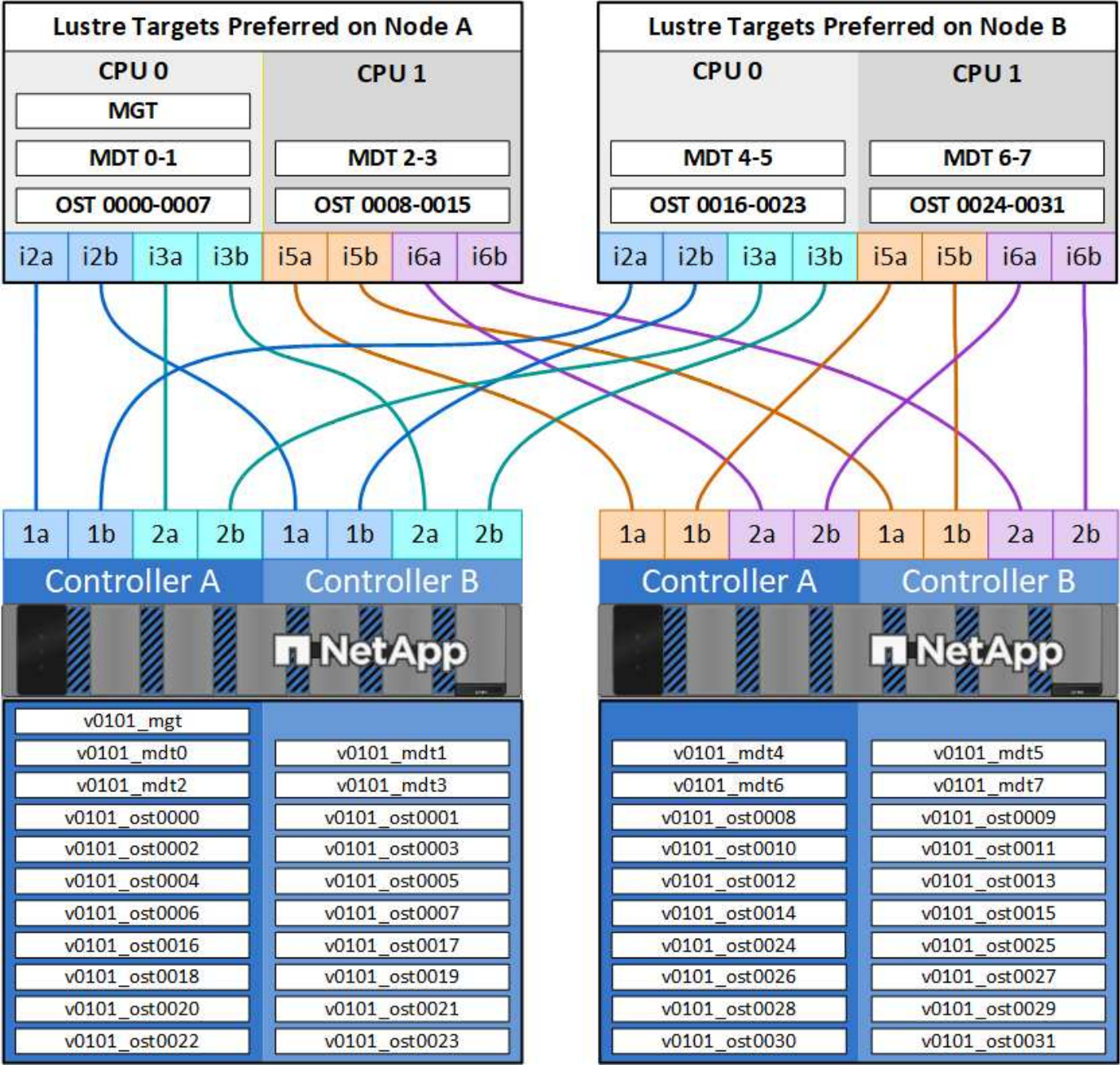
卷	建议大小	备注
MGS	5–10 GiB	仅配置数据
MDT	RAID 1 容量 ÷ MDT 数	每个约 1–2 TiB（典型值）
OST	RAID 6或DDP容量÷每个池的OST 计数	随驱动器容量扩展；请参见 "尺寸指南"

主要构建块卷分布

下图显示了基本 EF80 构建块中 OSS/MDS 服务器节点和 E-Series 阵列的首选 Lustre 目标放置和 NVMe-oF 连接。该图将管理目标标记为 **MGT**（管理目标）；一个 **MGT** 托管本文档其他地方引用的 **MGS**（管理服务器）服

务。

基础构建块目标分布（EF80）



下表列出了卷在两个阵列上的分布方式以及每个目标首选的服务器。

目标类型	阵列 1	阵列 2	服务器 1	服务器 2	备注
MGS	1	0	1	0	仅基础构建基块
MDT	4	4	4	4	每个服务器首选来自每个阵列的 2 个 MDT
OST	16	16	16	16	每个 RAID 6 池 8 个 × 每个阵列 2 个池，或等效的 DDP 分配

目标类型	阵列 1	阵列 2	服务器 1	服务器 2	备注
总卷数	21	20	21	20	

存储池选择：TLC 与 QLC

根据阵列中的 NVMe 驱动器容量选择 E-Series 池类型：

- **3.84 TB、7.68 TB 和 15.3 TB 驱动器：** 对 OST 卷使用 **RAID 6** 卷组，对 MGS 和 MDT 卷使用 **RAID 1** 卷组，或对共享驱动器池使用 DDP。
- **30.7 TB 和 61.4 TB 容量闪存 (QLC) 驱动器：** 仅使用 动态磁盘池 (DDP)。在 DDP 中为 MGS 和 MDT 存储创建 RAID 1 卷。从 DDP 为 OST 存储创建 RAID 6 卷。

有关按驱动器大小和布局的每个构建块可用容量估计，请参见 ["尺寸指南"](#)。

网络要求

后端(NVMe-oF)

- 每个 OSS/MDS 节点和每个 E-Series 阵列之间的 NVMe/InfiniBand 或 NVMe/RoCE
- NVMe/RoCE 后端接口上的 MTU 9000（典型值）
- 从每个 Lustre 节点到存储阵列的八个路径（每个阵列四个）
- EF80 每个节点使用六个 HCA（NVMe-oF 使用 i2、i3、i5 和 i6；LNet 使用 i1 和 i4）。将节点 A 的电缆连接到标签末尾为 a 的控制器端口，例如 `1a` 和 `2a`。将节点 B 的电缆连接到标签末尾为 b 的控制器端口，例如 `1b` 和 `2b`。请参阅 ["EF80 六 HCA 后端布线"](#)。

前端(LNet)

- IPoB 或 RoCE（LNet (@o2ib 网络类型）
- RoCE 前端接口上的 MTU 9000（典型值）
- 当有四个前端端口可用时，建议使用多轨 LNet（EF80：i1 和 i4 HCA，两个端口）
- 专用 InfiniBand 或 RoCE 交换机架构，连接 OSS/MDS 服务器节点和 Lustre 客户端
- 建议在 RoCE 网络中使用无损 RoCE（PFC）

管理

- 用于服务器 BMC 和阵列管理端口的带外管理网络
- 专用 Corosync 网络或共享前端结构（取决于站点）

使用 NetApp E 系列存储的 Lustre - 软件组件

查看使用 NetApp E-Series Storage 部署 Lustre 的软件堆栈和 Ansible 自动化。有关服务器、存储和网络，请参见 ["硬件组件"](#)。

构建块软件

下表列出了 E 系列阵列和 OSS/MDS 服务器节点使用的软件组件。使用 ["NetApp 互操作性矩阵工具 \(IMT\)"](#) 作为支持的版本和组件组合的权威来源。搜索 **E-Series Lustre** 并在部署之前确认操作系统、Lustre、SANtricity 操作系统、DOCA-OFED、HCA 固件和 HA 软件包版本。

组件	功能
SANtricity OS	在每个 E-Series 阵列上运行，并提供 SANtricity System Manager、具有自动 I/O 路径故障转移的双活动控制器高可用性、自动负载平衡和路径监控、动态磁盘池和传统 RAID、自动驱动器重建、镜像缓存保护、数据保证、主动驱动器运行状况监控以及在线软件、固件和配置更改。
Lustre	为客户端提供符合 POSIX 标准的单个命名空间，同时跨多个目标分发元数据和文件数据以进行并行访问。管理服务器 (MGS) 存储文件系统配置，元数据目标 (MDT) 管理命名空间，对象存储目标 (OST) 在 E-Series 卷上存储文件数据。这种分离可通过添加构建块来扩展性能和容量。
Red Hat Enterprise Linux (RHEL) 或 Rocky Linux	为 Lustre 目标服务、HA 集群、存储和网络连接提供内核、系统服务和包框架。受支持的操作系统存储库还提供 Pacemaker、Corosync、fence agents、NVMe-oF 和多路径软件包，这些软件包作为解决方案堆栈一起进行测试。
高可用性集群服务（Pacemaker、Corosync 和 fence agent）	这些服务共同为 OSS/MDS 服务器节点上的 Lustre 目标提供协调的高可用性。它们维护集群成员资格和仲裁，监控资源，协调有序故障转移，并在其他位置激活共享目标之前隔离无响应的节点。这种协调可防止对 E-Series 卷的脑裂访问，并在故障期间保护文件系统的完整性。
NVIDIA DOCA-OFED	为 InfiniBand 和 RoCE 结构提供 HCA 驱动程序、RDMA 库和管理实用程序。相同的软件堆栈支持对 E-Series 阵列的低延迟后端 NVMe-oF 访问以及与 Lustre 客户端的高吞吐量前端 LNet 通信。

NetApp 为 Rocky Linux 9.8 和 Red Hat Enterprise Linux 9.8 提供预构建的 Lustre 服务器 RPM。

Ansible 自动化

Lustre with NetApp E-Series Storage 通过控制节点使用 Ansible 自动化进行交付和部署，该控制节点具有对 E-Series 阵列和 OSS/MDS 服务器节点的管理访问权限。Ansible 清单模拟所需的解决方案，包括阵列、服务器、存储分配、网络接口和 HA 集群资源。

The NetApp E-Series Lustre Ansible 集合通过使用 SANtricity 和 Host 集合来协调端到端部署。这些集合一起配置和映射 E-Series 存储，准备服务器操作系统，配置 NVMe-oF 和 LNet 连接，部署 Lustre 目标，并创建 Pacemaker 资源。这种自动化使配置可重复，并简化了向现有文件系统添加构建基块的过程。

下表总结了 Ansible 控制节点上使用的主要软件包。有关当前软件包依赖项，请参见 ["NetApp E-Series Lustre Ansible 集合"](#)。

套餐	依赖关系或目的
Python	提供 Ansible 和所需 Python 模块的运行环境。
ansible-core	需要 Python 并运行 NetApp E-Series Ansible 集合。
其他 Python 包	cryptography、ipaddr、netaddr、passlib、resolvelib、jinja2 和 pyyaml

套餐	依赖关系或目的
NetApp E-Series Lustre Ansible 集合	配置 Lustre、LNet、NVMe-oF 主机连接和 Pacemaker 资源。
NetApp E-Series SANtricity Ansible 集合	通过 SANtricity Web Services API 配置 E 系列阵列、存储池、卷和主机映射。
NetApp E-Series 主机 Ansible 集合	为 LNet 和 NVMe-oF 与 E 系列阵列连接准备 OSS/MDS 服务器节点。

Lustre 客户端软件

Lustre 客户端软件使系统能够通过前端 LNet 结构访问并行文件系统。每个客户端都需要与其运行的内核和服务器的部署的 Lustre 版本兼容的 Lustre 内核模块和 Lustre 客户端实用程序。

使用["Lustre 支持表"](#)查找与 Lustre 版本兼容的客户端操作系统和内核。NetApp 不提供预构建的 Lustre 客户端包。从["netapp-lustre 存储库"](#)中 NetApp 提供的 Lustre 源代码为客户端操作系统和内核构建客户端包。安装 Lustre 客户端软件包后，`lustre\_client` 角色可以配置网络接口、LNet，并为 Lustre 文件系统创建持久的 systemd 挂载单元。

有关客户端部署步骤，请参见 ["部署解决方案"](#)。

基于 NetApp E 系列存储的 Lustre - 规模调整指南

根据容量和元数据需求调整 Lustre 与 NetApp E-Series 存储构建块的规模，决定何时增加容量，并审查影响性能的因素。

容量估算

包含两个 EF80 阵列的基础构建块可提供以下 Lustre 目标布局。有关首选目标位置和 NVMe-oF 路径的信息，请参见 ["硬件组件"](#) 中的 ["主要构建块卷分布"](#)。

组件	计数	典型卷大小（每个目标）
MGS	1	5-10 GiB
MDT	8	大小因驱动器容量和RAID布局而异
OST	32	大小因驱动器容量和RAID布局而异

每个 EF80 阵列使用 24 个 NVMe 驱动器。传统卷组（MGS/MDT 为 RAID 1，OST 为 RAID 6）或 DDP 支持的容量为 3.84 TB、7.68 TB 和 15.3 TB，仅 DDP 支持 30.7 TB 或 61.4 TB 容量闪存（QLC）驱动器。目前不建议将 1.92 TB 驱动器用于此解决方案。有关驱动器插槽布局和池选择，请参阅 ["硬件组件"](#)。

下表列出了 EF80 构建基块（两个阵列，48 个驱动器）的大致可用容量。阵列可用容量与最终 Lustre 文件系统可用容量不同。MDT inode 分配、日志、过度配置和库存量百分比会降低应用程序的可用容量。

驱动器容量	布局（每个数组）	大致可用性（构建基块）
3.84 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	138.08 TB
3.84 TB	DDP（24个驱动器，2个保留）	133.63 TB
7.68 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	276.35 TB

驱动器容量	布局（每个数组）	大致可用性（构建基块）
7.68 TB	DDP（24）	267.47 TB
15.3 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	552.88 TB
15.3 TB	DDP（24）	535.15 TB
30.7 TB	仅限 DDP	1,070.35 TB
61.4 TB	仅限 DDP	2,162.70 TB

分别调整元数据和数据的大小，然后在 Ansible 库存中设置卷大小。部署模板使用 `ldiskfs` 格式化目标；MDT 使用 Lustre 默认 inode 比率，OST 模板设置 `-i 4096`。

- **元数据（MDT）**：按预期文件计数调整 MDT 大小。规划大约为预期文件数量两倍的增长空间。MDT 过小可能会导致无法创建新文件，即使 OST 容量仍有剩余。
- **数据（OST）**：根据可用容量和吞吐量需求确定 OST 大小。
- **MGS**：仅对文件系统配置使用小管理目标（5-10 GiB）。

为所选驱动器容量调整清单中的 MDT 和 OST 卷大小。仅当站点需要不同的 inode 比率时，才更改 `format_options.mkfsoptions`` 中的 ``eseries_lustre_filesystem_mdt`` 或 ``eseries_lustre_filesystem_ost``。有关 inode 比率的背景知识，请参见["Lustre Wiki: Lustre Tuning"](#)。有关何时添加构建基块的信息，请参见[\[缩放\]](#)。

缩放

在容量或元数据负载需要时，添加构建基块：

- **仅限 OST**：文件计数和元数据加载在现有 MDT 容量范围内，您需要更多数据容量或聚合吞吐量。添加 32 个 OST 和两个 OSS/MDS 服务器节点。
- **MDT+OST**：文件计数或元数据速率接近 MDT 容量，或者您需要更多元数据吞吐量和数据容量。增加 8 个 MDT 和 32 个 OST。

在现有 MDT 上使用 ``mdt.*.md_stats`` 以确认元数据是否饱和。将每个 Pacemaker/Corosync 集群限制为五个构建块（十个 OSS/MDS 服务器节点）。对于较大的部署，将构建块均匀分布在多个 HA 集群中。有关扩展规则，请参阅["解决方案架构"](#)。

以下示例显示了一个 HA 集群中带有基本构建块和仅 OST 构建块的文件系统。

构建基块	类型	服务器	数组	MGS	MDT	OST
BB1	基础	2	2	1	8	32
BB2	仅限 OST	2	2	0	0	32
总计		4	4	1	8	64

BB2 使用 ``mgsnode=`` 针对 BB1 中的 MGS 注册其 OST 目标。BB2 的 OST 索引是全局分配的（例如，OST0032 到 OST0063），因此没有索引与 BB1 冲突。

性能

正式验证使用来自 Lustre 客户端的 IOR、mdtest 和 fio 来表征相对吞吐量、IOPS 和元数据行为，并验证 HA 故障转移。这些测试不会发布有保证的性能数据。综合基准测试衡量最佳情况下的行为，可能无法反映实际的应用程序性能。

性能大致随构建块的数量而扩展。实现的结果取决于驱动器类型和池布局、NVMe-oF 路径计数、LNet 结构和客户端计数，以及数据与元数据 I/O 的混合。

请联系您的 NetApp 客户团队，以获取特定于工作负载的规模和性能建议。

有关驱动器布局和卷计数，请参见["硬件组件"](#)。有关部署步骤，请参见["部署解决方案"](#)。有关现场级库存指导，请参见["自定义 Lustre Ansible 清单"](#)。

部署解决方案

使用 NetApp E-Series 存储部署 Lustre

通过按顺序完成硬件、环境准备、清单、Ansible 部署和验证任务，使用 NetApp E-Series 存储部署 Lustre。

前提条件

在开始部署之前，请确保满足以下条件：

- 您遵循了 ["尺寸指南"](#) 中的规格调整和构建块指南，并选择了与 ["硬件组件"](#) 匹配的硬件。
- 您确认服务器、HCA、E-Series、操作系统、Lustre、SANtricity OS 和 DOCA-OFED 组合已在 ["NetApp 互操作性表工具"](#) 的 **E-Series Lustre** 中列出。
- 计划构建模块的所有 OSS/MDS 服务器节点和 E-Series 阵列都已就位并准备上架。
- 您具有服务器 BMC、阵列管理端口以及将用作 Ansible 控制节点的主机的凭据和网络访问权限。



NetApp 支持使用 ["NetApp E-Series Lustre Ansible 集合"](#) 部署 Lustre HA 集群和 Lustre 客户端。请勿手动配置 Lustre 目标、LNet、NVMe-oF 主机连接或 Pacemaker 资源。

部署工作流

按顺序完成以下任务：

1. ["机架和线缆 Lustre 硬件"](#)。

架设 Lustre 服务器和 EF80 阵列，然后连接后端 NVMe-oF 和前端 LNet 结构。

2. ["准备 Lustre 部署环境"](#)。

配置阵列和服务器，然后在控制节点上安装 Ansible 及其依赖项。

3. ["自定义 Lustre Ansible 清单"](#)。

选择站点结构的模板，并填充服务器、阵列、网络、构建基块和凭据。

4. "部署 Lustre HA 集群和客户端"。

准备服务器操作系统、安装 NVIDIA DOCA-OFED、运行主部署 playbook，以及配置 Lustre 客户端。

5. "验证和扩展 Lustre 部署"。

在生产使用前确认集群运行状况、挂载和容量。当文件系统需要另一个构建基块时，扩展现有清单。

相关信息

- ["NetApp E-Series Lustre Ansible 集合"](#)
- ["NetApp Lustre 源库"](#)
- ["HA 管理用户指南"](#)
- ["性能调整指南"](#)
- ["Lustre 客户端部署模板"](#)

机架和布线 **Lustre** 硬件

架设 Lustre 服务器和 NetApp EF80 阵列，然后连接每个构建模块的后端 NVMe-oF 和前端 LNet 结构。

开始之前

查看 ["硬件组件"](#) 中的 HCA 计数、PCIe 插槽要求、路径要求和 MTU 指南。

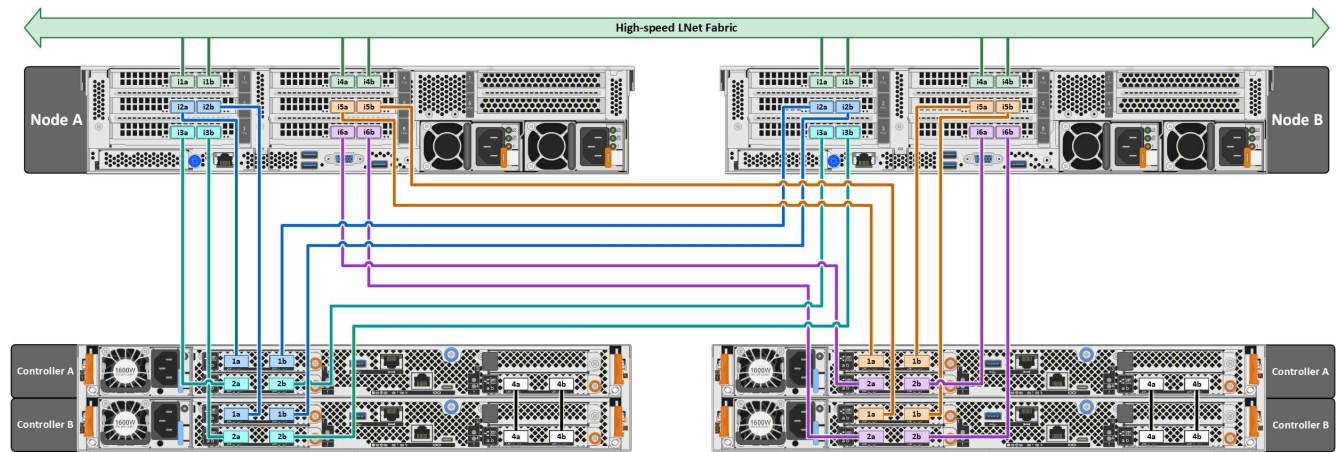
步骤

1. 根据构建块拓扑，对所有 Lustre 服务器和 E-Series 存储阵列进行机架安装和线缆连接。



2. 对于每个 EF80 阵列，将控制器 A 端口 4a 连接到控制器 B 端口 4a，将控制器 A 端口 4b 连接到控制器 B 端口 4b。这些专用连接提供控制器间镜像，不用于主机流量。请参阅 ["用缆线连接 EF50 和 EF80 控制器间镜像连接"](#)。
3. 根据 EF80 六 HCA 拓扑，在服务器和阵列之间连接后端 NVMe-oF 网络。

EF80 六 HCA 后端布线 (RoCE 或 InfiniBand)



4. 将每个 Lustre 服务器和 Lustre 客户端上的四个前端 LNet 接口 (i1 和 i4 HCA) 连接到专用 InfiniBand 或 RoCE 交换机网络。
5. 使用 RoCE 时, 使用优先级流量控制 (PFC) 配置网络结构以实现无损操作。Ansible playbooks 在构建基块的所有接口上配置无损 RoCE。

完成后

"准备服务器、阵列和 Ansible 控制节点"。

准备 Lustre 部署环境

在每个 EF80 阵列和 Lustre 服务器上配置管理访问和基本设置, 然后准备 Ansible 控制节点以部署解决方案。

准备 E 系列阵列

步骤

1. 在每个控制器上配置管理接口, 并验证 SANtricity System Manager 是否可访问。
2. 设置 Ansible 清单将使用的阵列名称和管理凭据。
3. 确认阵列处于最佳状态, 并且所有驱动器均存在且报告正常。

有关配置管理连接的说明, 请参见 "[E-Series 文档](#)"。

准备 Lustre 服务器

步骤

1. 在每台服务器上配置基板管理控制器 (BMC) 以进行带外访问, 并设置 UEFI 系统设置以实现最高性能。
2. 安装受支持的操作系统 (RHEL 或 Rocky Linux), 然后配置主机名和管理网络端口。
3. 准备所有固件和软件包, 以满足 **E-Series Lustre** 解决方案的要求, 如 "[软件组件](#)" 中所述。

准备 Ansible 控制节点

指定具有对所有 Lustre 服务器和 E-Series 阵列的管理网络访问权限的虚拟或物理 Linux 系统作为 Ansible 控制

节点。

以下步骤使用 Ubuntu 24.04。有关其他 Linux 发行版的说明，请参见 ["Ansible 安装文档"](#)。

步骤

1. 安装 Python、pip 和 Python 虚拟环境包：

```
sudo apt-get install python3 python3-pip python3-setuptools python3-venv
```

2. 创建并激活 Python 虚拟环境：

```
python3 -m venv ~/pyenv  
source ~/pyenv/bin/activate
```

3. 在激活的虚拟环境中安装 Ansible 和所需的 Python 包：

```
pip install "ansible-core>=2.19" cryptography jinja2 ipaddr netaddr \  
passlib pyyaml resolvelib
```

4. 安装 NetApp E-Series Lustre Ansible collection。collection 安装还会安装其依赖 collection，包括 SANtricity 和 Host collection：

```
ansible-galaxy collection install netapp_eseries.lustre
```

5. 验证已安装的 Ansible、Python 和集合版本：

```
ansible --version  
python3 --version  
ansible-galaxy collection list netapp_eseries.lustre
```

6. 配置从控制节点到每个 Lustre 服务器的无密码 SSH。将 <ip_or_hostname> 替换为服务器的管理 IP 地址或主机名：

```
ssh-keygen  
ssh-copy-id <ip_or_hostname>
```

对每个 Lustre 服务器重复 `ssh-copy-id` 上述步骤。



请勿为 E 系列阵列配置无密码 SSH。Ansible 使用清单中定义的凭据，通过 SANtricity Web Services API 管理阵列。

完成后

"自定义 Ansible 清单"。

自定义 Lustre Ansible 清单

创建 EF80 部署模板的工作副本，并为服务器、阵列、网络 and Lustre 构建基块自定义其清单。

将选定的清单保留在其 fabric 和平台目录下。保留 playbooks 根目录下的 shared 目录和 passwords.yml 文件 building_blocks。

选择部署模板

从与站点的网络结构匹配的模板开始：

- **InfiniBand:** ["IB_H2_IB_DAC_A2_EF80/lustre_building_block_cluster_1"](#)
- **RoCE:** ["ROCE_H2_ROCE_DAC_A2_EF80/lustre_building_block_cluster_1"](#)

配置特定于站点的设置

在清单中设置站点特定的值，以及本地存储库和 udev 规则设置。表中的行引用指向 ansible-lustre release/1.0.0 标记中的 RoCE EF80 模板文件。InfiniBand 模板使用相同的变量名称和结构，但 LNet 接口变量除外，该变量在表中注明。

字段	Ansible 变量	模板文件 (release/1.0.0)	范围	备注
文件系统名称	eseries_lustre_filesystem_mgt. format_options. fsname, eseries_lustre_filesystem_mdt. format_options. fsname, eseries_lustre_filesystem_ost. format_options. fsname	"ha_cluster.yml#L76" , "#L90" , "#L110"	集群范围	嵌套在 MGT、MDT 和 OST 映射下的 format_options。最多 8 个字符。在所有三个位置使用相同的值。顶级 `fsname` 键无效。
用于目标注册的 MGS NIDs	eseries_lustre_filesystem_mdt. format_options. mgsnode, eseries_lustre_filesystem_ost. format_options. mgsnode	"ha_cluster.yml#L98" , "#L118"	集群范围	嵌套在 MDT 和 OST 下 format_options。包括 HA 对中两个节点的前端 LNet NID。首先列出配置为首选 MGS 所有者的节点的 NID，然后列出合作伙伴节点的 NID。

字段	Ansible 变量	模板文件 (release/1.0.0)	范围	备注
Pacemaker 集群名称	eseries_lustre_pacemaker_cib_properties_overrides.cluster_properties.cluster-name	"ha_cluster.yml#L228"	集群范围	嵌套在 Pacemaker CIB 属性覆盖下。HA 集群的唯一名称。
BMC 隔离地址	eseries_lustre_pacemaker_fencing_agents.fence_redfish.ip	"ha_cluster.yml#L211", "#L214", "#L217", "#L220"	每个 Lustre 服务器重复	每个服务器节点的 Redfish BMC IP 地址。
警报电子邮件收件人	eseries_lustre_alert_email_list	"ha_cluster.yml#L233"	集群范围	HA 资源和故障通知的收件人。
邮件域	eseries_lustre_mail_service_overrides.conf.mydomain	"ha_cluster.yml#L240"	集群范围	用于发送警报电子邮件的后缀域。
NTP 时间源	eseries_lustre_time_sync_overrides.server_pools	"ha_cluster.yml#L250"	集群范围	集群节点的时间源。指定一个 NTP 池或一个或多个单独的 NTP 服务器。包括所需的 pool 或 server 指令和选项，例如 iburst。
接口 PCI 到名称 udev 映射	eseries_ip_udev_rules	"ha_cluster.yml#L172"	每个 HCA 插槽布局重复	将每个 PCI 插槽映射到持久接口名称 (L182 处的值)。使用 lspci -nnvmm 收集。
本地包存储库 (可选)	eseries_common_yum_repos	"ha_cluster.yml#L256", "#L261"	集群范围	对于气隙安装，请取消注释该块并设置存储库 url (L264)。
服务器管理 IP	ansible_host	"host_vars/lustre_01.yml#L10"	每个 Lustre 服务器重复	每台服务器一个 host_vars/lustre_NN.yml 文件。
NVMe-oF 后端接口	eseries_nvme_roce_interfaces (RoCE) / eseries_nvme_ib_interfaces (InfiniBand)	"RoCE host_vars/lustre_01.yml#L21"、"InfiniBand host_vars/lustre_01.yml#L17"	每个 Lustre 服务器重复	用于连接 E 系列阵列的 NVMe-oF 存储流量的后端接口地址。
LNet 集群接口	eseries_roce_interfaces (RoCE) / eseries_ipoib_interfaces (InfiniBand)	"RoCE host_vars/lustre_01.yml#L47"、"InfiniBand host_vars/lustre_01.yml#L38"	每个 Lustre 服务器重复	前端 LNet 接口地址 (RoCE 值在 L56; InfiniBand 值在 L47)。

字段	Ansible 变量	模板文件 (release/1.0.0)	范围	备注
Corosync 节点 IP	eseries_lustre_corosync_node_ips	"host_vars/lustre_01.yml#L58"	每个 Lustre 服务器重复	列出 HA 集群中的所有节点 IP (L62 处的值)。
Lustre 目标服务 NID	lustre_target_service_nodes	"host_vars/lustre_01.yml#L64"	每个 Lustre 服务器重复	包括 HA 对中两个节点的前端 LNet NID (@o2ib)，以便在故障转移后仍可访问目标服务 (L67 处的值)。
阵列管理 IP	ansible_host	"host_vars/netapp_01.yml#L13"	每个 E 系列阵列重复	一个控制器的管理 IP；API URL 从中派生。
构建基块组成员身份	mgs / mds_NN / oss_NN	"lustre_inventory.yml"	每个构建基块	每个库存主机名必须与其相应的 host_vars`文件的基本名称匹配。例如，主机`lustre_01`使用`host_vars/lustre_01.yml`，而 netapp_01`使用`host_vars/netapp_01.yml`。仅在第一个构建基块中包含 MGS 组。在基本构建基块和每个 MDT+OST 扩展构建基块中包含 MDS 组；仅从纯 OST 扩展构建基块中省略 MDS 组。使用`mgsnode`将扩展目标注册到现有 MGS。

重复每个主机和构建基块的设置。每个 Lustre 服务器创建一个`host\_vars/lustre\_NN.yml`文件，每个阵列创建一个`host\_vars/netapp\_NN.yml`文件，并为每个额外的构建基块重复清单组。

部署 playbook 从共享`building\_blocks/passwords.yml`文件加载阵列、Pacemaker 和 BMC 凭据。在部署之前填充此文件并使用 Ansible Vault 对其进行加密，或在运行时使用`--extra-vars`安全地覆盖值。请勿将纯文本凭据提交到源代码管理。



部署模板中的 IP 地址、主机名、接口名和卷大小均为示例。在运行任何 playbook 之前，请修改环境的所有清单文件。

完成后

"部署 Lustre HA 集群和客户端"。

部署 Lustre HA 集群和客户端

使用 NetApp Ansible playbook 准备服务器操作系统、安装 NVIDIA DOCA-OFED、部署 Lustre HA 集群，并配置 Lustre 客户端。

在开始之前，请填写并固定 "自定义 Lustre Ansible 清单" 中所述的清单。

部署 Lustre HA 集群

步骤

1. 从工作副本中的 `building\_blocks/playbooks` 目录，在 Lustre 服务器上准备操作系统。将 `` 替换为自定义清单目录的路径：

```
ansible-playbook -i <inventory_path>/lustre_inventory.yml
deploy_lustre_os/deploy_lustre_os_playbook.yml
```

2. 在每台 Lustre 服务器上安装 NVIDIA DOCA-OFED。根据上一步安装的内核构建内核模块。按照 ["NVIDIA DOCA-Host 安装和升级"](#) 中适用于您内核的操作步骤进行。以下示例显示了在 RHEL 或 Rocky Linux 上安装 DOCA-OFED 的过程。
 - a. 安装 DOCA 主机存储库包并刷新包缓存。将 `` 替换为适用于您的操作系统和架构的 DOCA 主机包：

```
dnf install -y wget tar
wget https://www.mellanox.com/downloads/DOCA/<doca_host_package>.rpm
rpm -i <doca_host_package>.rpm
dnf clean all
dnf makecache
```

- b. 为正在运行的内核构建 DOCA 内核模块。脚本将打印其创建的软件包的路径：

```
dnf install -y doca-extra
/opt/mellanox/doca/tools/doca-kernel-support
```

- c. 安装脚本创建的内核模块存储库包，然后刷新包缓存。将 `` 替换为上一步中的路径：

```
rpm -Uvh <doca_kernel_repo>.rpm
dnf makecache
```

- d. 安装 DOCA-OFED 用户空间、内核和 NVMe 包。将 `` 替换为正在运行的内核版本：

```
dnf install -y doca-ofed-userspace
dnf install -y --disablerepo=doca doca-kernel-<kernel_version>
dnf install -y kmod-mlnx-nvme
```

3. 运行主要部署 playbook：

```
ansible-playbook -i <inventory_path>/lustre_inventory.yml
```

```
lustre_playbook.yml
```



调整 `--forks`` 以适应部署规模，从而控制 Ansible 并行配置的主机数量。例如，对于较大的集群，请添加 ``--forks 20`。

此 Playbook 负责配置 E-Series 卷组或 DDP 池和卷、配置 NVMe-oF 主机连接、格式化和挂载 Lustre 目标、配置 LNet、应用性能调优，以及配置 Pacemaker HA 资源。



一个包含一个构建块的部署构成一个双节点 Pacemaker 集群。在故障转移的情况下，节点 A 会获得额外投票。要在双节点集群中实现仲裁，请考虑配置 `qdevice`。当部署包含多个构建块时，每个构建块为较大的 Pacemaker 集群提供一个双节点服务器对，最高可达已记录的集群限制。查看 ["HA 管理用户指南"](#) 中的隔离和仲裁配置，以便集群可以在节点故障期间安全地恢复目标。

部署 Lustre 客户端

通过自定义客户端部署模板并运行客户端 playbook，在任何需要文件系统访问权限的系统上部署 Lustre 客户端。

步骤

1. 从 ["Lustre 支持表"](#) 中选择兼容的客户端操作系统和内核。如果尚未安装，请从 ["netapp-lustre"](#) 为该内核构建并安装 Lustre 客户端软件包。
2. 创建 ``clients`` 模板目录的工作副本，包括其共享 ``passwords.yml`` 文件，然后选择与您的架构匹配的模板：
 - **InfiniBand**: ["IB_lustre_cluster_clients"](#)
 - **RoCE**: ["ROCE_lustre_cluster_clients"](#)
3. 为您的客户端自定义 `lustre_client_inventory.yml`、`host_vars` 和 `group_vars`。表中的行引用指向 `ansible-lustre release/1.0.0` 标记中的 RoCE 客户端模板文件；除另有说明外，InfiniBand 客户端模板使用相同的变量。

字段	Ansible 变量	模板文件 (release/1.0.0)	范围	备注
客户端主机	<code>lustre_clients</code>	"lustre_client_inventory.yml#L4"	每位客户	列出每个客户端主机名。
客户端 SSH 用户和提权密码	<code>ssh_client_user</code> / <code>ssh_client_become_pass</code>	"passwords.yml#L2"	集群范围	仅当 SSH 用户不是 <code>`root`</code> 时，才设置提权密码。
LNet接口	<code>eseries_lustre_lnet</code>	"group_vars/all.yml#L7"	集群范围	客户端 LNet 接口名称（L13 处的值）。
挂载 MGS NIDs	<code>lustre_client_mounts.mgsnode</code>	"group_vars/all.yml#L17"	集群范围	用于挂载文件系统的 MGS NID（L20 的值）。
文件系统名称	<code>lustre_client_mounts.fsname</code>	"group_vars/all.yml#L21"	集群范围	必须与集群 <code>`fsname`</code> 匹配。

字段	Ansible 变量	模板文件 (release/1.0.0)	范围	备注
挂载点	lustre_client_mounts.mount_point	"group_vars/all.yml#L22"	集群范围	客户端挂载目录。
客户端管理 IP	ansible_host	"host_vars/client_01.yml#L4"	每个客户端重复	每个客户端一个 `host_vars/client_NN.yml` 文件。
存储结构接口	eseries_roce_interfaces	"host_vars/client_01.yml#L10"	每个客户端重复	前端接口地址（L15 处的值）。InfiniBand 模板在此处使用 eseries_ipoib_interfaces。

为每个客户端重复设置。为每个客户端创建一个 `host_vars/client_NN.yml` 文件，并将相应的主机添加到 `lustre_client_inventory.yml`。填充共享 `clients/passwords.yml` 文件，并在运行客户端 playbook 之前使用 Ansible Vault 对其进行加密。

4. 从客户端模板目录运行客户端 playbook:

```
ansible-playbook -i lustre_client_inventory.yml
lustre_client_playbook.yml
```

客户端 playbook 配置客户端网络接口和 LNet，可以为 Lustre 文件系统创建一个持久的 systemd 挂载单元。

完成后

"验证部署"。

验证和扩展 Lustre 部署

在生产使用之前确认集群运行状况、Lustre 挂载和文件系统容量，然后在需要时使用相同的清单添加构建块。

验证部署

步骤

1. 在 Lustre 服务器上，确认所有 Pacemaker 资源均已在预期节点上启动：

```
pcs status
```

2. 从 Lustre 客户端，确认 Lustre 文件系统已挂载：

```
mount -t lustre
```

3. 从 Lustre 客户端，确认文件系统容量以及 MDT 和 OST 目标可见：

```
lfs df -h
```

有关使用 IOR、mdtest 和 fio 进行性能验证，请参见 ["尺寸指南"](#)。

对于受控目标迁移或 HA 故障转移测试，请按照 ["HA 管理用户指南"](#) 中的步骤进行操作。

扩展文件系统

要增加容量或元数据性能，请使用额外的构建块扩展现有库存。请勿为新构建块创建单独的库存。该手册自动分配唯一的 MDT 和 OST 索引，并根据现有 MGS 注册新目标。

步骤

1. 将新的构建基块主机、阵列和资源组（MDT+OST 或仅限 OST）添加到用于原始部署的相同清单以及 `host\_vars` 和 `group\_vars` 文件中。
2. 根据更新的清单重新运行主要部署 playbook。

有关扩展规则，请参阅 ["解决方案架构"](#)；有关何时添加每种构建基块类型的指导，请参阅 ["尺寸指南"](#)。

使用 NetApp E-Series 存储的 Lustre - 相关资源

在规划或扩展 Lustre with NetApp E-Series Storage 部署时，请使用这些链接获取相关文档、工具和软件。有关调整大小和部署步骤，请参阅 ["尺寸指南"](#) 和 ["部署解决方案"](#)。

NetApp 资源

- ["NetApp 互操作性表工具"](#)。搜索 **E-Series Lustre**。
- ["NetApp EF-Series 全闪存阵列数据表"](#)
- ["NetApp SANtricity System Manager 文档"](#)
- ["NetApp ansible-lustre 集合"](#)。有关清单字段指南，请参见 ["自定义 Lustre Ansible 清单"](#)。
- ["NetApp Lustre 源库"](#)。NetApp 为 Rocky Linux 9.8 和 Red Hat Enterprise Linux 9.8 提供预构建的 Lustre 服务器 RPM。从 NetApp 源代码为客户端操作系统和内核构建 Lustre 客户端包。

Lustre 社区

- ["Lustre 文件系统"](#)
- ["Lustre 支持表"](#) 适用于客户端操作系统和内核
- ["Lustre Wiki: Lustre Tuning"](#)

相关 NetApp AI 基础设施解决方案

- ["NetApp 上的 BeeGFS 与 E 系列存储"](#)
- ["利用 NetApp E-Series 存储部署 IBM Spectrum Scale"](#)

- "搭载NetApp EF 系列的NVIDIA DGX SuperPOD"

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。