



NetApp E-Series 스토리지를 사용하는 **Lustre** NetApp artificial intelligence solutions

NetApp
August 31, 2026

목차

NetApp E-Series 스토리지를 사용하는 Lustre	1
Lustre with NetApp E-Series 스토리지 - 소개	1
솔루션 개요	1
Lustre with NetApp E-Series 스토리지 - 솔루션 아키텍처	1
빌딩 블록 디자인	2
확장성 권장 사항	3
HA 아키텍처	4
네트워크 아키텍처	5
Lustre 클라이언트	5
Lustre와 NetApp E-Series 스토리지 - 하드웨어 구성 요소	6
서버 요구 사항	6
EF80 표준 구성 요소	8
스토리지 풀 선택: TLC vs QLC	12
네트워크 요구 사항	12
Lustre with NetApp E-Series 스토리지 - 소프트웨어 구성 요소	12
빌딩 블록 소프트웨어	12
Ansible 자동화	13
Lustre 클라이언트 소프트웨어	14
Lustre with NetApp E-Series 스토리지 - 사이징 가이드	14
용량 사이징	14
확장	15
성능	16
솔루션 배포	16
NetApp E-Series 스토리지를 사용하여 Lustre 배포	16
랙 및 케이블 Lustre 하드웨어	17
Lustre 배포 환경 준비	18
Lustre Ansible 인벤토리를 사용자 지정합니다	20
Lustre HA 클러스터 및 클라이언트 배포	23
Lustre 배포 확인 및 확장	26
Lustre with NetApp E-Series 스토리지 - 관련 자료	27
NetApp 리소스	27
Lustre 커뮤니티	27
관련 NetApp AI 인프라 솔루션	28

NetApp E-Series 스토리지를 사용하는 Lustre

Lustre with NetApp E-Series 스토리지 - 소개

NetApp E-Series 스토리지를 사용하는 Lustre는 AI 및 HPC 워크로드를 위한 검증된 병렬 파일 시스템 솔루션으로, HA 서버 쌍과 E-Series 올플래시 어레이라는 반복 가능한 구성 요소로 이루어져 있습니다.

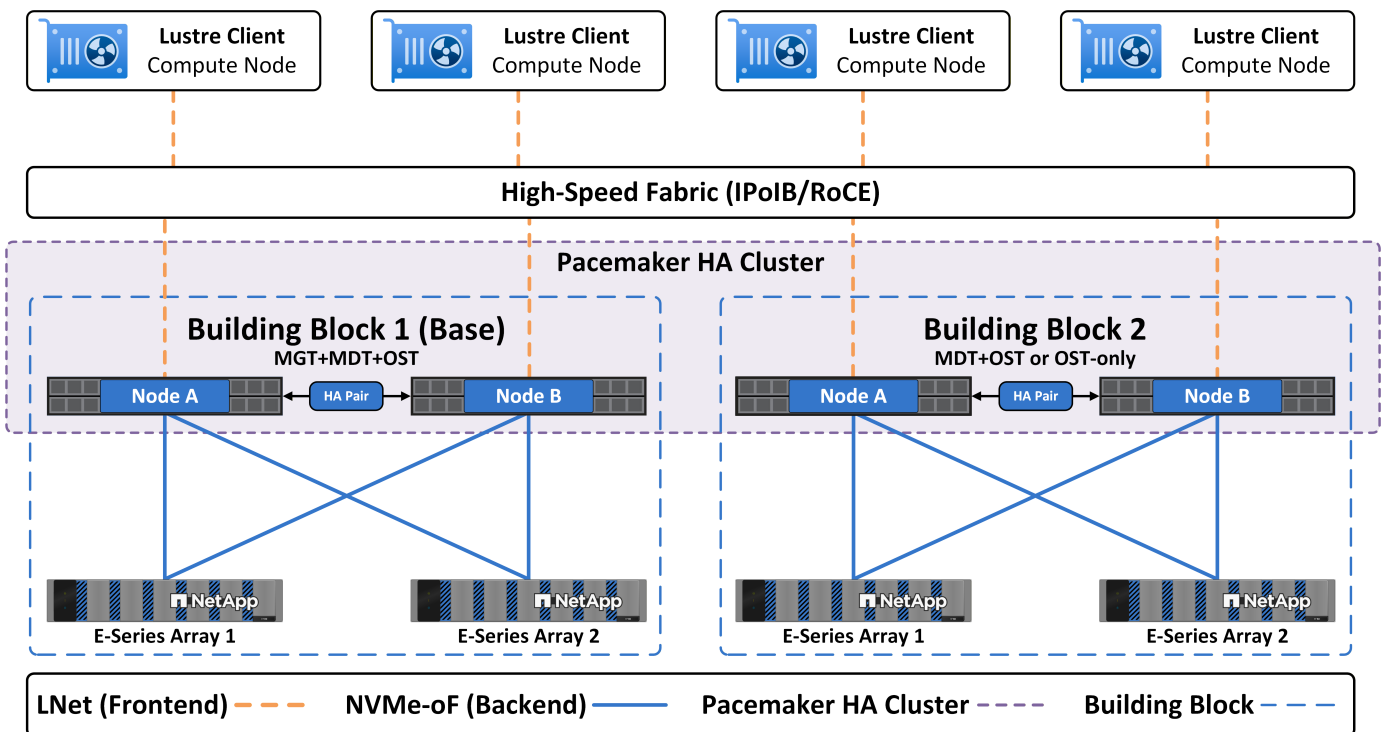
솔루션 개요

Lustre와 NetApp E-Series 스토리지는 오픈 소스 Lustre 병렬 파일 시스템과 NetApp E-Series 블록 스토리지를 결합하여 데이터 집약적인 워크로드를 위한 확장 가능하고 고성능의 기반을 제공합니다. 각 구성 요소는고가용성 (HA)으로 구성된 두 개의 OSS/MDS 서버 노드와 두 개의 E-Series 올플래시 어레이로 이루어져 있습니다. 백엔드 NVMe-oF 연결을 통해 블록 I/O가 어레이로 전달되며, 별도의 LNet 패브릭을 통해 클라이언트와 OSS/MDS 서버 노드를 연결하여 병렬 파일 시스템에 액세스할 수 있습니다.

기본 구성 요소는 관리 서버(MGS)와 초기 메타데이터 대상(MDT) 및 오브젝트 스토리지 대상(OST)을 호스팅합니다. 추가 구성 요소는 기본 MGS에 등록되어 워크로드 요구 사항이 증가함에 따라 메타데이터 용량, 데이터 용량 및 전체 처리량을 확장합니다.

다음 그림은 LNet을 통해 연결된 여러 구성 요소와 Lustre 클라이언트를 포함하는 배포 예시를 보여줍니다.

Lustre with NetApp E-Series 스토리지 솔루션 개요



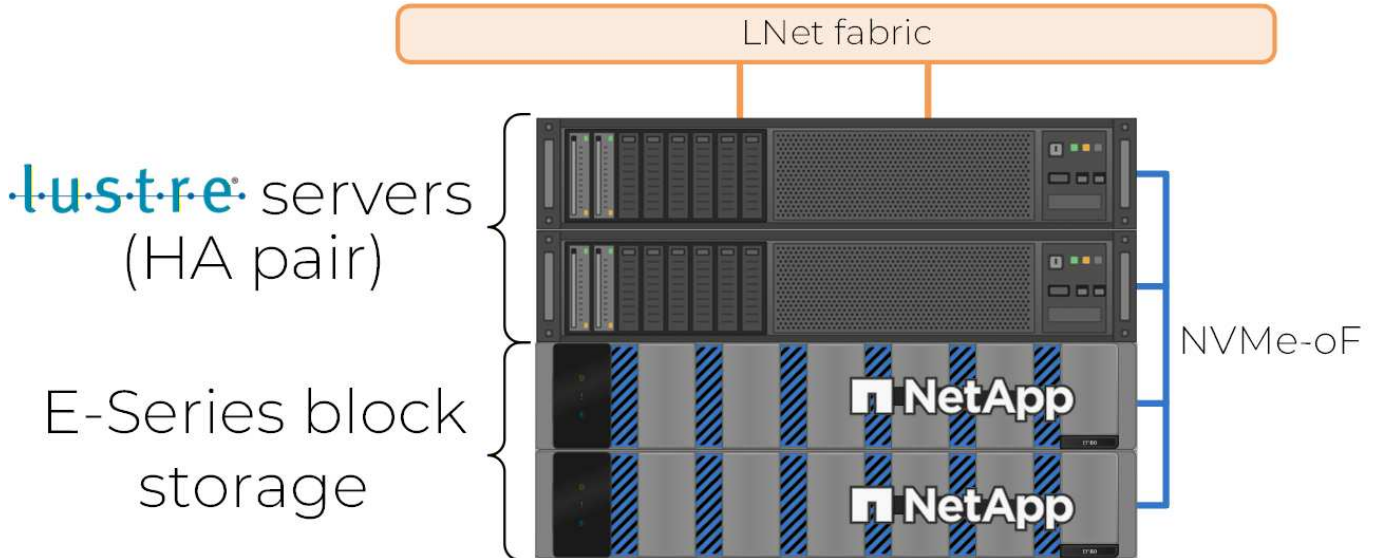
Lustre with NetApp E-Series 스토리지 - 솔루션 아키텍처

NetApp E-Series 스토리지를 사용하는 Lustre가 구성 요소,고가용성, 네트워크 토폴로지 및 클라이언트를 활용하여 용량, 성능 및 장애 조치를 계획하는 방법을 알아보십시오.

빌딩 블록 디자인

빌딩 블록은 NetApp E-Series Lustre 솔루션의 기본 확장 단위입니다. 각 빌딩 블록은 Lustre 오브젝트 스토리지 서버(OSS) 및 메타데이터 서버(MDS) 기능을 제공하는 고가용성(HA) 쌍으로 구성된 두 개의 서버 노드, 두 개의 NetApp E-Series 올플래시 어레이, 서버와 어레이 간의 전용 백엔드 NVMe over Fabrics(NVMe-oF) 연결, 그리고 클라이언트와 노드 간 통신을 위한 전용 프런트엔드 Lustre 네트워킹(LNet) 연결로 구성됩니다. Pacemaker 및 Corosync HA 클러스터는 하나 이상의 빌딩 블록에서 서버 쌍을 포함할 수 있습니다. NetApp ansible-lustre 컬렉션은 Ansible 자동화를 사용하여 Lustre 서비스를 배포하고 구성합니다.

Lustre 빌딩 블록



빌딩 블록은 파일 시스템의 성장 요구 사항에 맞춰 확장됩니다. 모든 파일 시스템에는 하나의 기본 빌딩 블록이 필요합니다. 기본 빌딩 블록은 관리 대상(MGT)에 관리 서버(MGS)를 호스팅하고 초기 메타데이터 대상(MDT) 및 오브젝트 스토리지 대상(OST) 용량을 제공합니다. 추가 빌딩 블록은 파일 시스템을 확장하고 기본 빌딩 블록의 MGS에 MDT 및 OST 대상을 등록합니다. 이러한 모듈식 접근 방식을 통해 워크로드 요구 사항에 따라 용량과 성능을 독립적으로 확장할 수 있습니다.

NetApp은 대부분의 배포 환경에 대해 다음과 같은 빌딩 블록 구성 프로파일을 권장합니다. 나열된 MGS, MDT 및 OST 개수는 E-Series 스토리지 어레이의 성능과 용량을 극대화하도록 최적화된 권장 기본값입니다. 워크로드 요구 사항에 맞게 Ansible 인벤토리에서 대상 개수, 볼륨 크기 및 E-Series 드라이브 할당을 조정하십시오. 예를 들어, 더 많은 메타데이터 스토리지 용량이 필요한 배포 환경에서는 동일한 빌딩 블록 하드웨어 내에서 더 많은 MDT와 더 적은 OST를 프로비저닝할 수 있습니다.

다음 표는 구성 요소 유형과 권장 목표 개수를 요약한 것입니다.

유형	MGS	MDTs	OST	사용
베이스	1	8	32	파일 시스템의 첫 번째 구성 요소입니다. MGS와 초기 MDT 및 OST 용량을 호스팅합니다.
MDT+OST	0	8	32	메타데이터 및 데이터 용량을 추가합니다. 기본 구성 요소인 MGS에 등록하십시오.

유형	MGS	MDTs	OST	사용
OST 전용	0	0	32	데이터 용량만 추가합니다. 기본 구성 요소인 MGS에 등록하십시오.

- 드라이브 유형 및 풀 레이아웃: E-Series는 기존 RAID 볼륨 그룹과 동적 디스크 풀(DDP)을 지원합니다. TLC NVMe 드라이브의 경우, OST 스토리지에는 RAID 6 볼륨 그룹을, MGS 및 MDT 스토리지에는 RAID 1 볼륨 그룹을 사용하십시오. QLC NVMe 드라이브의 경우, 공유 드라이브 풀에 DDP를 사용하십시오.
- 타겟 분산: 각 빌딩 블록의 Lustre 타겟은 OSS/MDS 서버 노드와 E-Series 어레이 모두에 걸쳐 균형 있게 분산됩니다. 이러한 레이아웃은 서버 CPU와 어레이 컨트롤러에 I/O를 분산시켜 NVMe-oF 경로 성능을 향상시키고 이중화를 유지합니다. "[타겟 분포 및 NVMe-oF 연결성](#)"의 "[하드웨어 구성 요소](#)"를 참조하십시오.

지원되는 운영 체제, Lustre 버전, 어레이 펌웨어 및 관련 구성 요소는 "[NetApp 상호 운용성 매트릭스 툴\(IMT\)](#)" **E-Series Lustre** 항목에 나열되어 있습니다. 자세한 볼륨 수, 드라이브 레이아웃 및 크기 조정 지침은 "[하드웨어 구성 요소](#)"를 참조하십시오. 소프트웨어 구성 요소는 "[소프트웨어 구성 요소](#)"를 참조하십시오.

확장성 권장 사항

Lustre 파일 시스템은 메타데이터 용량, 데이터 용량 또는 둘 다 제한 요소가 될 때 구성 요소를 추가하여 확장합니다. MDT는 파일 수와 메타데이터 IOPS를 기준으로 확장하고, OST는 용량 및 총 처리량 요구 사항을 기준으로 확장합니다.

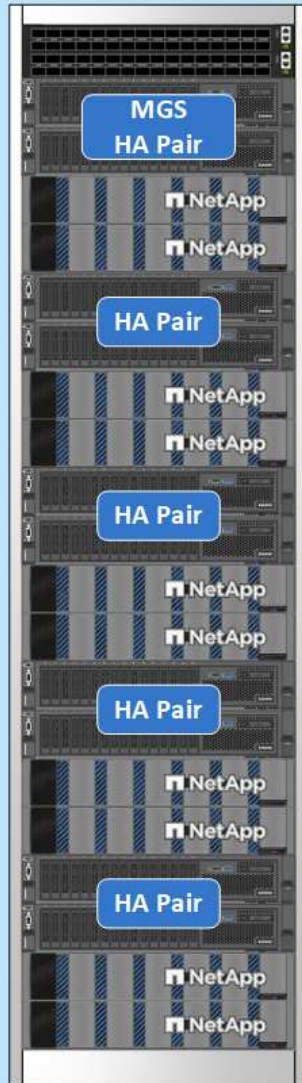
1. 파일 시스템당 하나의 기본 빌딩 블록: 기본 빌딩 블록을 정확히 하나만 배포하십시오. 이 블록에는 MGS와 초기 MDT 및 OST 대상이 포함됩니다.
2. 추가 빌딩 블록 프로필: 기존 MDT 용량이 충분하고 데이터 용량 또는 처리량을 늘려야 하는 경우 OST 전용 빌딩 블록을 추가하십시오. 메타데이터 성능 또는 네임스페이스 용량이 병목 현상이 되면 MDT+OST 빌딩 블록을 추가하십시오.
3. **HA** 클러스터 제한: 각 Pacemaker 및 Corosync HA 클러스터는 최대 5개의 빌딩 블록(10개의 Lustre 서버 노드)으로 제한됩니다. 대규모 배포 환경에서는 리소스 제약을 방지하기 위해 여러 HA 클러스터로 균등하게 분할하십시오.

다음 그림은 42U 랙에 최대 5개의 빌딩 블록이 쌓인 모습을 보여줍니다(랙당 Pacemaker HA 클러스터 1개). 여러 랙이 하나의 Lustre 파일 시스템에 참여할 수 있으며, 기본 빌딩 블록에만 MGS가 호스팅됩니다.

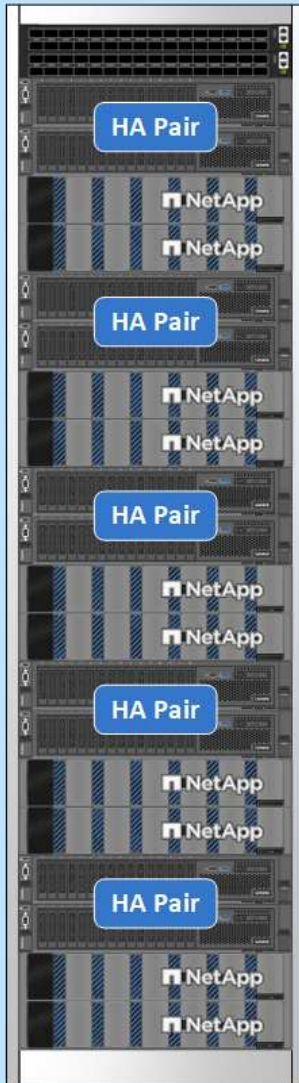
러스터 병렬 파일 시스템 랙 스케일링

Lustre Parallel Filesystem

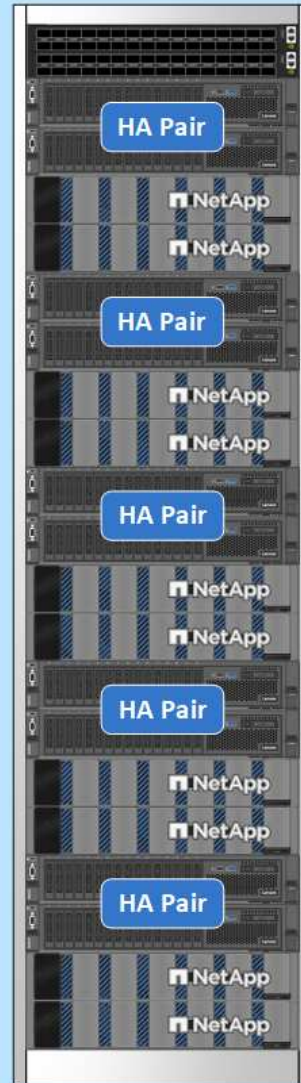
Standalone
HA Cluster



Standalone
HA Cluster



Standalone
HA Cluster



사이즈 예시는 "[사이즈 가이드](#)"을 참조하십시오.

HA 아키텍처

Lustre with NetApp E-Series 스토리지는 NetApp E-Series 스토리지와 통합된 공유 디스크고가용성(HA) 아키텍처를 사용합니다. Pacemaker는 HA 리소스를 관리하고 Corosync는 클러스터 멤버십 및 메시징을 제공합니다. 이 둘은 함께 각 빌딩 블록의 두 OSS/MDS 서버 노드 간 Lustre 타겟 페일오버를 관리합니다. 다중 경로 NVMe-oF는 두 OSS/MDS 서버 노드를 동일한 E-Series 볼륨에 연결하므로 필요에 따라 어느 노드든 타겟 서비스를 인계받을 수 있습니다.

각 MGS, MDT 및 OST는 종속 리소스와 함께 Pacemaker 리소스 그룹에 HA 리소스로 구성됩니다. Pacemaker는 리소스가 올바른 순서로 시작 및 중지되고, 동일한 노드에 함께 배치되며, 한 번에 하나의 노드에서만 실행되도록 보장합니다. 두 노드에서 동일한 대상에 동시에 액세스하면 기본 파일 시스템이 손상될 수 있습니다.

- **타겟 페일오버:** OSS/MDS 서버 노드는 클러스터에서 서로 동등한 위치에 있지만, 각 Lustre 타겟은 한 번에 하나의 노드에서만 활성화됩니다. Pacemaker 모니터링 리소스는 각 타겟과 해당 종속 요소의 상태를 감시하고, 타겟이 현재 노드에서 사용 불가능해지면 페일오버를 트리거합니다. 장애 발생 시 Pacemaker는 파트너 노드에서 영향을 받는 리소스 그룹을 올바른 순서로 다시 시작하고, 서비스가 재개되면 클라이언트는 새 위치의 타겟에 투명하게 다시 연결됩니다. 각 타겟이 단일 노드에서만 활성화되므로 파일 시스템은 두 노드에서 동시에 쓰기 작업에 노출되지 않습니다.
- **공유 스토리지 액세스:** 다중 경로 NVMe-oF 경로는 각 OSS/MDS 서버 노드를 빌딩 블록의 모든 E-Series 컨트롤러에 연결하여 모든 타겟 볼륨에 어느 노드에서든 접근할 수 있도록 합니다. 서버 노드에 장애가 발생하면 파트너 노드는 이미 동일한 볼륨에 대한 활성 경로를 보유하고 있으므로 스토리지 연결을 재구성하지 않고도 타겟에 대한 소유권을 인계받을 수 있습니다. 어레이 컨트롤러 또는 경로에 장애가 발생하면 다중 경로가 I/O를 정상 작동 중인 컨트롤러로 리디렉션합니다. 이러한 이중 중복성을 통해 Lustre 타겟은 백업 볼륨에 대한 액세스를 잃지 않고 OSS/MDS 서버 노드 또는 어레이 컨트롤러 간에 장애 조치를 수행할 수 있습니다.
- **STONITH 펜싱:** 장애가 발생하면 Pacemaker는 때때로 장애가 발생한 노드와 통신하여 해당 노드의 타겟이 중지되었는지 확인할 수 없습니다. 이러한 타겟을 다른 곳에서 재시작하기 전에 Pacemaker는 전원을 차단하는 등의 방법으로 장애가 발생한 노드를 펜싱하여 완전히 다운되었는지 확인합니다. 펜싱은 두 노드가 동시에 동일한 타겟에 액세스하여 기본 파일 시스템을 손상시키는 스플릿 브레인 시나리오를 방지하고, 장애 조치 중에 데이터 무결성을 유지합니다. NetApp은 Redfish를 지원하는 베이스보드 관리 컨트롤러(BMC)가 있는 서버에 `fence_redfish``를 권장합니다. 다른 펜싱 에이전트 (``fence_apc`` 등)도 지원됩니다.

클러스터 관리 및 펜싱 구성에 대해서는 "[HA 사용자 가이드](#)"의 "[ansible-lustre 컬렉션](#)"을 참조하십시오.

네트워크 아키텍처

이 솔루션은 스토리지 백엔드 트래픽과 LNet 프런트엔드 트래픽에 대해 별도의 네트워크 경로를 사용합니다.

- **백엔드(NVMe-oF):** OSS/MDS 서버 노드는 NVMe/InfiniBand 또는 NVMe/RoCE를 사용하여 NVMe-oF를 통해 E-Series 어레이에 연결됩니다. NVMe-oF 경로는 Lustre 타겟 서비스와 E-Series 볼륨 간에 블록 I/O를 전송합니다. EF80 6-HCA 백엔드 케이블 연결은 "[Lustre 하드웨어 랙 및 케이블 연결](#)"을 참조하십시오. 노드 및 어레이 전반의 기본 타겟 배치에 대해서는 "[하드웨어 구성 요소](#)"의 "[목표 분포](#)"를 참조하십시오.
- **프런트엔드(LNet):** Lustre 클라이언트와 OSS/MDS 서버 노드는 NVIDIA OFED 스택을 사용하는 @o2ib 네트워크 유형인 LNet을 통해 통신합니다. InfiniBand와 RoCE는 모두 검증된 프런트엔드 전송 프로토콜입니다. `ansible-lustre` 컬렉션은 모든 프런트엔드 인터페이스를 멀티레일 LNet으로 구성하여 여러 포트를 통합함으로써 대역폭을 높이고 클라이언트 및 노드 간 트래픽에 대한 경로 이중화를 제공합니다.
- **관리 및 클러스터:** 아웃오브밴드(out-of-band) 관리 네트워크는 서버 관리, BMC 액세스 및 어레이 관리를 제공합니다. STONITH 펜싱 에이전트 등 ``fence_redfish``은 이 네트워크를 사용하여 서버 BMC에 연결하고 장애가 발생한 노드의 전원을 차단합니다. Corosync는 프런트엔드 패브릭과 함께 추가 링으로 이 네트워크를 통해 클러스터 통신을 실행할 수도 있습니다. Corosync를 여러 링으로 구성하면 클러스터 메시징에 대한 이중화가 추가되어 단일 네트워크 장애로 인해 쿼럼이 중단될 위험이 줄어듭니다.

Lustre 클라이언트

Lustre 클라이언트는 클라이언트 커널 모듈을 로드하고, OSS/MDS 서버 노드와의 LNet 연결을 설정한 후, 파일 시스템을 마운트하여 애플리케이션에 단일하고 일관된 POSIX 호환 네임스페이스를 제공합니다. I/O는 활성 OST 전체에 병렬로 분산됩니다. 클라이언트 노드는 빌딩 블록 외부에 있으며 프런트엔드 LNet 패브릭을 통해 파일 시스템을 사용합니다. 이 솔루션에 대한 클라이언트 운영 체제는 IMT에 나열되어 있지 않습니다. 배포된 Lustre 릴리스에서 지원하는 테스트된 클라이언트 배포판 및 커널 버전(예: Red Hat Enterprise Linux 9, SUSE Linux Enterprise Server 15, Ubuntu 24.04)은 "[Lustre 지원 매트릭스](#)"를 참조하십시오.

클라이언트 노드는 OSS/MDS 서버 노드와 동일한 LNet 패브릭 및 네트워크 유형에 연결되어야 합니다. 필요한 경우 LNet 라우터를 사용하여 서로 다른 LNet 서브넷 또는 패브릭을 연결할 수 있습니다.

NetApp은 Rocky Linux 9.8 및 Red Hat Enterprise Linux 9.8용 Lustre 서버 RPM 패키지를 제공합니다. NetApp은 클라이언트 패키지를 제공하지 않습니다. "[netapp-lustre repository](#)"의 Lustre 소스 코드에서 클라이언트 운영 체제 및 커널용 클라이언트 패키지를 빌드하십시오.

클라이언트 패키지가 설치되면, 선택적 `lustre_client` 역할이 "[ansible-lustre 컬렉션](#)" 클라이언트 네트워크 인터페이스와 LNet을 구성하고, 선택 시 멀티레일 LNet을 활성화하고, 연결성을 검증하고, 영구적인 `systemd` 마운트 장치를 관리합니다. 이 역할은 Lustre를 빌드하지 않습니다. `lustre_client_packages`가 채워진 경우에만 클라이언트 패키지를 설치합니다. 필요한 경우 클라이언트를 수동으로 구성할 수 있습니다. 단계별 절차는 "솔루션 배포" 및 "lustre_client 역할 문서"를 참조하십시오.`

Lustre와 NetApp E-Series 스토리지 - 하드웨어 구성 요소

서버, NetApp E-Series 어레이, 스토리지 레이아웃 및 네트워킹에 대한 이 하드웨어 요구 사항은 NetApp E-Series 스토리지를 사용하여 Lustre를 구성하고 주문할 때 활용하십시오. 소프트웨어 구성 요소에 대한 자세한 내용은 "[소프트웨어 구성 요소](#)"를 참조하십시오.

서버 요구 사항

이 솔루션은 BYOS(Bring Your Own Server) 모델을 사용합니다. 각 구성 요소에는 다음 사양을 충족하거나 초과하는 두 개의 OSS/MDS 서버 노드가 필요합니다.

노드 사양

다음 표는 OSS/MDS 노드별 최소 서버 요구 사항을 나열합니다.

요소	요구 사항
수량	건물 블록당 노드 2개
CPU	AMD EPYC 또는 Intel Xeon, 32코어 이상
메모리	256 GB DDR5 (최소)
네트워크 HCAs	6개의 듀얼 포트 200Gb HCA(HCA 연결성 참조)
PCIe 슬롯	2× PCIe Gen5 x16 및 4× PCIe Gen5 x8 (PCIe 슬롯 요구 사항 참조)
부팅 드라이브	RAID 1 구성의 드라이브 2개 (소프트웨어 또는 하드웨어 RAID 권장)

NetApp은 Lenovo ThinkSystem SR665 V3 서버를 사용하여 솔루션을 검증했습니다. 다른 공급업체의 동등한 서버도 이러한 요구 사항을 충족하는 경우 지원됩니다.

HCA 연결성

다음 표는 OSS/MDS 서버 노드별 HCA, LNet 프론트엔드 포트 및 NVMe-oF 경로 요구 사항을 나열합니다.

노드당 HCA	Lustre 노드당 LNet 포트	Lustre 노드당 NVMe-oF 경로	예시 HCA
6개 (12개 포트)	4	총 8개 (각 배열에 4개씩)	MCX755106AS-HEAT (듀얼 포트 200Gb PCIe gen5 카드)

NetApp은 EF80 스토리지 어레이의 대역폭을 극대화하기 위해 노드당 6개의 HCA를 권장합니다.

PCIe 슬롯 요구 사항

EF80 빌딩 블록의 각 OSS/MDS 서버 노드에는 6개의 듀얼 포트 HCA가 있습니다. 이 중 2개는 LNet용이고 4개는 NVMe-oF용입니다. 슬롯 너비는 각 HCA에서 사용할 수 있는 대역폭을 결정하므로 카드 자체의 너비가 아닌 모든 슬롯과 라이저의 전기적 너비를 확인해야 합니다. Gen5 x16 카드를 Gen5 x8 슬롯에 설치하면 x8 속도로 작동합니다.

- **LNet HCAs:** 프런트엔드 처리량을 극대화하려면 PCIe Gen5 x16 슬롯에 설치하십시오.
- **NVMe-oF HCAs:** PCIe Gen5 x8 또는 PCIe Gen5 x16 슬롯에 설치하십시오.
- **NUMA 균형:** HCA를 두 NUMA 영역에 균등하게 분배하여 각 영역에 LNet 인터페이스 2개와 NVMe-oF 인터페이스 4개가 할당되도록 합니다.

다음 표는 EF80 빌딩 블록에서 각 OSS/MDS 서버 노드에 필요한 최소 슬롯 수를 나타냅니다.

트래픽 유형	노드당 HCA	최소 슬롯 폭	NUMA 구역별 슬롯
LNet	2	PCIe Gen5 x16	1
NVMe-oF	4	PCIe Gen5 x8	2

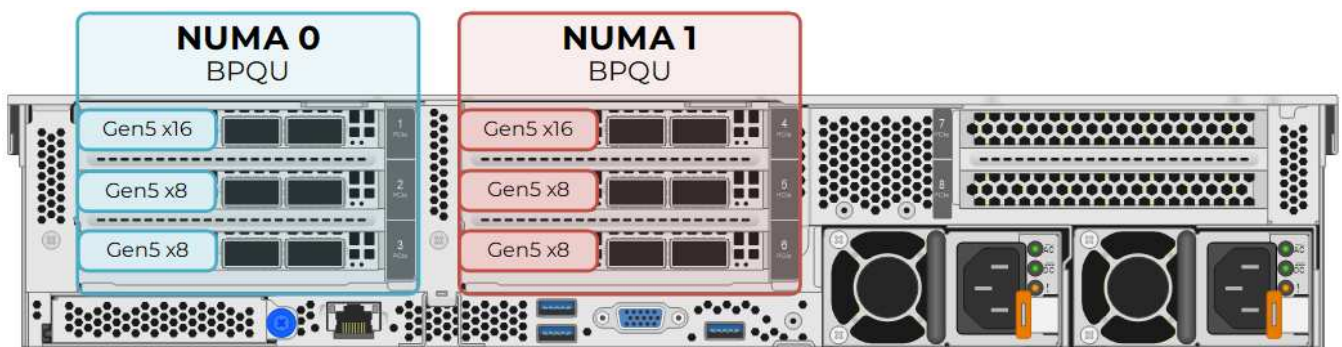
선택적으로 듀얼 포트 HCA에서 포트를 분할하여 한 포트는 LNet 트래픽을, 다른 포트는 NVMe-oF 트래픽을 처리하도록 할 수 있습니다. 모든 LNet 포트가 최대 처리량을 발휘하도록 하려면 이 HCA 4개를 모두 PCIe Gen5 x16 슬롯에 설치한 다음, 나머지 NVMe-oF 포트를 위해 Gen5 x8 또는 Gen5 x16 슬롯에 HCA 2개를 추가하십시오.

▼ Lenovo ThinkSystem SR665 V3 라이저 구성 예

다음 두 가지 라이저 조합 모두 EF80 빌딩 블록에 필요한 슬롯 요구 사항을 충족합니다.

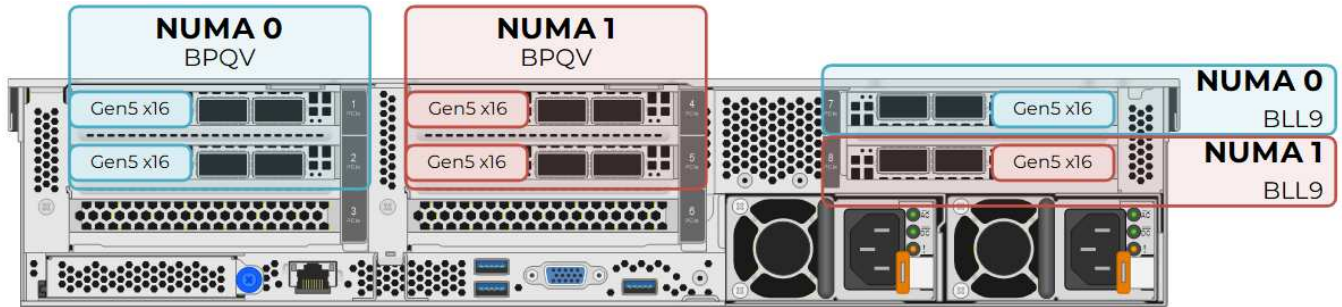
최소 슬롯 너비

BPQU 라이저를 라이저 위치 1과 2에 설치하십시오. 각 BPQU 라이저는 PCIe Gen5 x16 슬롯 1개와 PCIe Gen5 x8 슬롯 2개를 제공합니다. 이 라이저들을 함께 사용하면 LNet용 Gen5 x16 슬롯 2개와 NVMe-oF용 Gen5 x8 슬롯 4개를 NUMA 영역에 균등하게 분배하여 사용할 수 있습니다.



모든 Gen5 x16 슬롯

라이저 위치 1과 2에 BPQV 라이저를, 라이저 위치 3에 BLL9 라이저를 설치하십시오. 각 BPQV 라이저는 두 개의 PCIe Gen5 x16 슬롯을 제공합니다. BLL9 라이저는 NUMA 영역 0에 슬롯 7, NUMA 영역 1에 슬롯 8을 제공하며, 두 슬롯 모두 PCIe Gen5 x16입니다. 이 조합을 통해 NUMA 영역당 세 개씩 총 여섯 개의 Gen5 x16 슬롯을 사용할 수 있으며, 동일한 HCA의 포트 간에 LNet 및 NVMe-oF 트래픽을 분산할 수 있습니다.



EF80 표준 구성 요소

EF80은 이번 솔루션 릴리스를 위한 검증된 표준 플랫폼입니다.

어레이 사양

다음 표는 빌딩 블록(두 개의 어레이)에 대한 EF80 어레이 사양을 나열합니다.

요소	사양
모델	NetApp EF80
폼 팩터	2U 기본 새시, 24개의 내부 NVMe SSD 슬롯
컨트롤러	듀얼 컨트롤러(A 및 B)
드라이브	어레이당 24개의 NVMe SSD
입출력 연결성	검증된 Lustre 설계에서 어레이당 8× 200Gb NVMe/IB 또는 NVMe/RoCE 호스트 포트

EF80 플랫폼은 각 컨트롤러에 2포트 호스트 I/O 모듈 3개를 설치할 경우 어레이당 최대 12개의 호스트 포트를 지원합니다. 검증된 Lustre 설계는 슬롯 1과 2에 호스트 I/O 모듈을 사용하여 어레이당 8개의 호스트 포트를 제공합니다.

각 EF80 어레이는 컨트롤러 간 미러링을 위해 전용 슬롯 4 I/O 모듈을 사용합니다. 컨트롤러 A의 포트 4a를 컨트롤러 B의 포트 4a에, 컨트롤러 A의 포트 4b를 컨트롤러 B의 포트 4b에 연결하십시오. 이러한 연결은 캐시 미러링 및 I/O 전송을 제공하며 호스트 NVMe-oF 트래픽에는 사용되지 않습니다. 을 참조하십시오. ["EF50 및 EF80 컨트롤러 간 미러링 연결 케이블 연결"](#)

모든 모델의 EF-Series 전체 사양은 ["NetApp EF 시리즈 올플래시 어레이 데이터시트"](#)를 참조하십시오.

드라이브 레이아웃 (어레이당 24개 드라이브)

기본 구성 요소의 각 EF80 어레이는 24개의 NVMe 드라이브 슬롯을 모두 사용합니다.

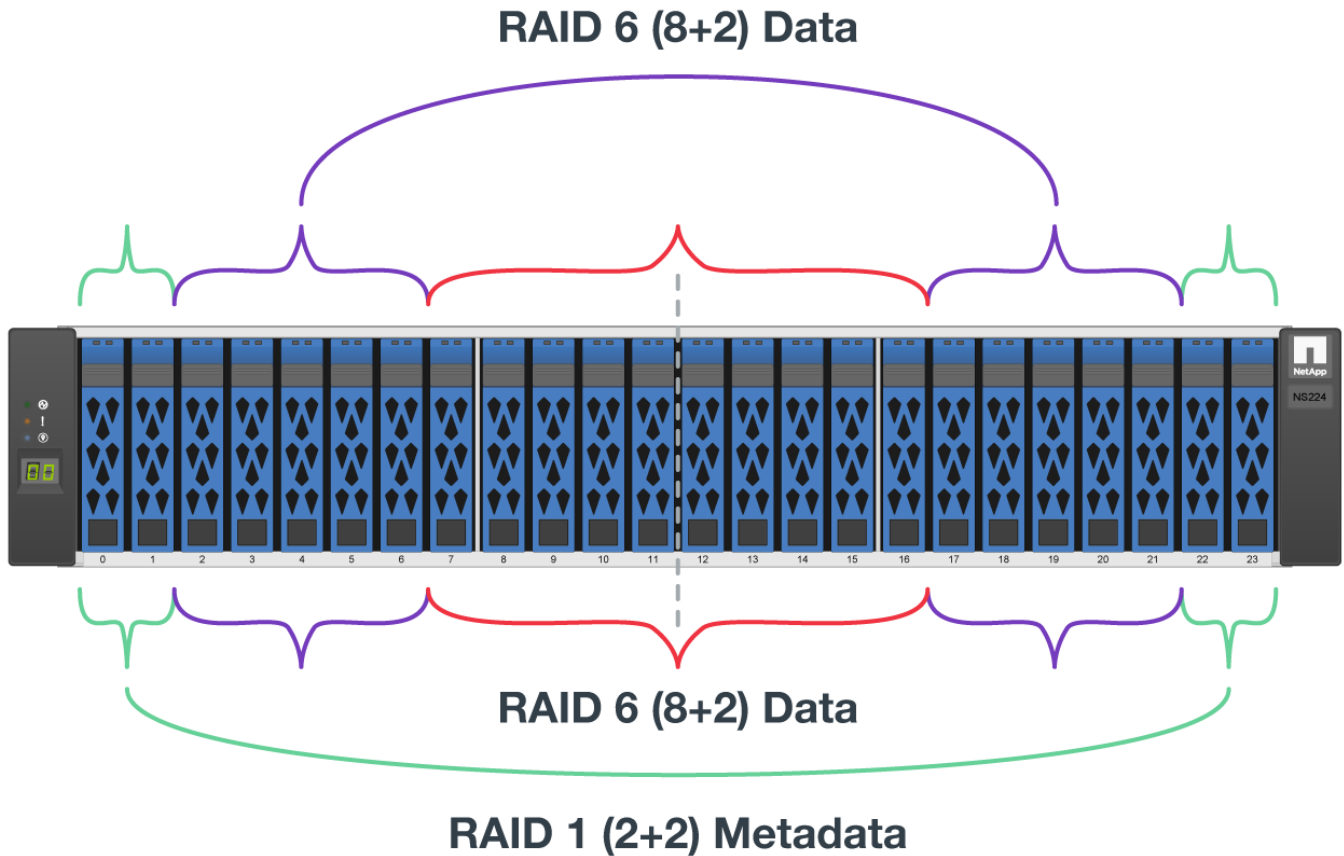
- MGS/MDT 저장소용 RAID 1 구성의 드라이브 4개 (공유 볼륨 그룹 또는 DDP 할당)

- OST 저장소(첫 번째 OST 풀)용 RAID 6 구성의 드라이브 10개
- OST 저장소(두 번째 OST 풀)용 RAID 6 구성의 드라이브 10개

이 레이아웃은 기존 볼륨 그룹에서 3.84TB, 7.68TB 또는 15.3TB 드라이브를 사용할 때 적용됩니다. 30.7TB 또는 61.4TB Capacity Flash(QLC) 드라이브를 사용하는 경우, 24개 드라이브 전체에 걸쳐 단일 Dynamic Disk Pool을 프로비저닝하고, MGS 및 MDT 볼륨에는 풀 내부에 RAID 1 볼륨을 생성하고, OST에는 기본 RAID 6 볼륨을 사용하십시오.



1.92TB 드라이브는 현재 이 솔루션에 권장되지 않습니다. 위에 나열된 검증된 드라이브 용량 중 하나를 사용하십시오.



볼륨 및 타겟 수(기본 빌딩 블록)

다음 표는 기본 구성 요소에 대한 MGS, MDT 및 OST 수를 나열합니다.

대상 유형	BB당 개수	RAID/풀	참고
MGS	1	RAID 1	기본 구성 요소만 해당; 배열 1
MDT	8	RAID 1	배열당 4개
OST	32	RAID 6(TLC) 또는 DDP(QLC)	배열당 16개

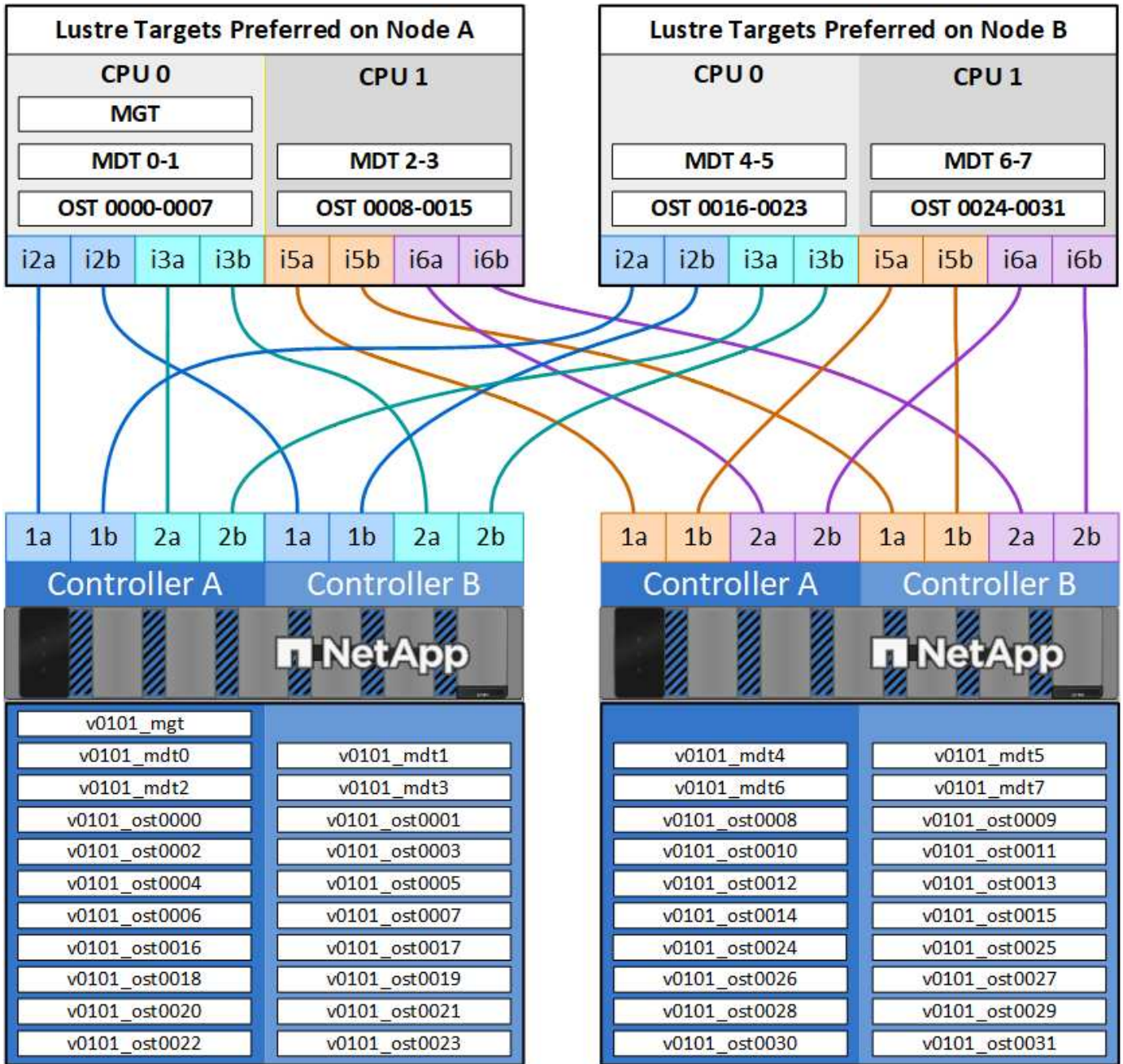
Ansible 인벤토리의 시작점으로 다음 볼륨 사이징 지침을 사용하십시오.

용량	권장 크기	참고
MGS	5–10 GiB	구성 데이터만
MDT	RAID 1 용량 ÷ MDT 개수	개당 약 1~2 TiB (일반적)
OST	RAID 6 또는 DDP 용량 ÷ 풀당 OST 개수	드라이브 용량에 따라 확장됩니다. 자세한 내용은 " 사이즈 가이드 "을 참조하십시오.

기본 구성 요소 볼륨 배포

다음 그림은 기본 EF80 빌딩 블록에서 OSS/MDS 서버 노드와 E-Series 어레이 간의 Lustre 타겟 배치 및 NVMe-oF 연결 구성을 보여줍니다. 다이어그램에서 관리 타겟은 **MGT**(관리 타겟)로 표시되어 있으며, 하나의 MGT는 이 문서의 다른 부분에서 참조되는 MGS(관리 서버) 서비스를 호스팅합니다.

기본 구성 요소 목표 분포(EF80)



다음 표는 두 어레이에 걸쳐 볼륨이 어떻게 분산되는지, 그리고 각 대상이 어떤 서버를 선호하는지를 보여줍니다.

대상 유형	어레이 1	어레이 2	Server 1	서버 2	참고
MGS	1	0	1	0	기본 빌딩 블록만 해당
MDT	4	4	4	4	각 서버는 각 어레이에서 2개의 MDT를 선호합니다.
OST	16	16	16	16	RAID 6 풀당 8개 × 어레이당 2개 풀 또는 이에 상응하는 DDP 할당
총 볼륨	21	20	21	20	

스토리지 풀 선택: TLC vs QLC

어레이의 NVMe 드라이브 용량에 따라 E-Series 풀 유형을 선택하십시오.

- **3.84TB, 7.68TB 및 15.3TB** 드라이브: OST 볼륨에는 *RAID 6 볼륨 그룹*을, MGS 및 MDT 볼륨에는 *RAID 1 볼륨 그룹*을 사용하거나 공유 드라이브 풀에 DDP를 사용하십시오.
- **30.7TB 및 61.4TB** 용량의 플래시(QLC) 드라이브: *동적 디스크 풀(DDP)*만 사용하십시오. MGS 및 MDT 스토리지를 위해 DDP 내부에 RAID 1 볼륨을 생성하십시오. OST 스토리지를 위해 DDP에서 RAID 6 볼륨을 생성하십시오.

빌딩 블록별 드라이브 크기 및 레이아웃에 따른 사용 가능 용량 추정치는 "[사이즈 가이드](#)"를 참조하십시오.

네트워크 요구 사항

백엔드(NVMe-oF)

- 각 OSS/MDS 노드와 각 E-Series 어레이 간의 NVMe/InfiniBand 또는 NVMe/RoCE
- NVMe/RoCE 백엔드 인터페이스의 MTU 9000(일반)
- 각 Lustre 노드에서 스토리지 어레이로 연결되는 경로는 총 8개입니다(어레이당 4개).
- EF80은 노드당 6개의 HCA(NVMe-oF의 경우 i2, i3, i5 및 i6, LNet의 경우 i1 및 i4)를 사용합니다. 노드 A의 케이블을 레이블이 a`로 끝나는 컨트롤러 포트(예: `1a 및 2a)에 연결합니다. 노드 B의 케이블을 레이블이 b`로 끝나는 컨트롤러 포트(예: `1b 및 2b)에 연결합니다. "[EF80 6-HCA 백엔드 케이블링](#)"을 참조하십시오.

프런트엔드(LNet)

- LNet (@o2ib 네트워크 유형의 경우 IPoIB 또는 RoCE)
- RoCE 프런트엔드 인터페이스의 MTU 9000 (일반적인 경우)
- 프런트엔드 포트가 4개 사용 가능한 경우(EF80: i1 및 i4 HCA, 두 포트 모두) 멀티레일 LNet 사용을 권장합니다.
- OSS/MDS 서버 노드와 Lustre 클라이언트를 연결하는 전용 InfiniBand 또는 RoCE 스위치 패브릭
- RoCE 패브릭에는 무손실 RoCE(PFC)를 권장합니다.

관리

- 서버 BMC 및 어레이 관리 포트용 대역 외 관리 네트워크
- 전용 Corosync 네트워크 또는 공유 프런트엔드 패브릭(사이트에 따라 다름)

Lustre with NetApp E-Series 스토리지 - 소프트웨어 구성 요소

NetApp E-Series 스토리지를 사용하여 Lustre를 배포하는 소프트웨어 스택 및 Ansible 자동화를 검토하십시오. 서버, 스토리지 및 네트워크에 대한 자세한 내용은 "[하드웨어 구성 요소](#)"를 참조하십시오.

빌딩 블록 소프트웨어

다음 표는 E-Series 어레이 및 OSS/MDS 서버 노드에서 사용되는 소프트웨어 구성 요소를 나열합니다. "[NetApp 상호](#)

운용성 매트릭스 툴(IMT)"를 지원되는 버전 및 구성 요소 조합에 대한 권위 있는 소스로 사용하십시오. *E-Series Lustre*를 검색하여 배포 전에 운영 체제, Lustre, SANtricity OS, DOCA-OFED, HCA 펌웨어 및 HA 패키지 버전을 확인하십시오.

요소	기능
SANtricity OS	각 E-Series 어레이에서 실행되며 SANtricity System Manager, 자동 I/O 경로 페일오버를 통한 듀얼 액티브 컨트롤러 고가용성, 자동 로드 밸런싱 및 경로 모니터링, 동적 디스크 풀 및 기존 RAID, 자동 드라이브 재구축, 미러링된 캐시 보호, Data Assurance, 사전 예방적 드라이브 상태 모니터링, 온라인 소프트웨어, 펌웨어 및 구성 변경 기능을 제공합니다.
Lustre	클라이언트에게 단일 POSIX 호환 네임스페이스를 제공하는 동시에 메타데이터와 파일 데이터를 여러 대상에 분산하여 병렬 액세스를 지원합니다. 관리 서버 (MGS)는 파일 시스템 구성을 저장하고, 메타데이터 대상(MDT)은 네임스페이스를 관리하며, 오브젝트 스토리지 대상(OST)은 E-Series 볼륨에 파일 데이터를 저장합니다. 이러한 분리를 통해 구성 요소를 추가하여 성능과 용량을 확장할 수 있습니다.
Red Hat Enterprise Linux(RHEL) 또는 Rocky Linux	Lustre 타겟 서비스, 고가용성 클러스터링, 스토리지 및 네트워크 연결을 위한 커널, 시스템 서비스 및 패키지 프레임워크를 제공합니다. 지원되는 운영 체제 리포지토리에는 Pacemaker, Corosync, 펜스 에이전트, NVMe-oF 및 다중 경로 패키지도 포함되어 있으며, 이러한 패키지들은 솔루션 스택으로 함께 테스트되었습니다.
고가용성 클러스터 서비스(Pacemaker, Corosync 및 fence agents)	이러한 서비스들은 함께 OSS/MDS 서버 노드 전반에 걸쳐 Lustre 타겟에 대한 고가용성을 제공합니다. 클러스터 멤버십 및 쿼럼을 유지하고, 리소스를 모니터링하며, 순차적인 페일오버를 오케스트레이션하고, 응답하지 않는 노드를 격리한 후 다른 곳에서 공유 타겟을 활성화합니다. 이러한 조정을 통해 E-Series 볼륨에 대한 스플릿 브레인 액세스를 방지하고 장애 발생 시 파일 시스템 무결성을 보호합니다.
NVIDIA DOCA-OFED	InfiniBand 및 RoCE 패브릭용 HCA 드라이버, RDMA 라이브러리 및 관리 유틸리티를 제공합니다. 동일한 소프트웨어 스택은 E-Series 어레이에 대한 저지연 백엔드 NVMe-oF 액세스와 Lustre 클라이언트와의 고처리량 프론트엔드 LNet 통신을 지원합니다.

NetApp 은 Rocky Linux 9.8 및 Red Hat Enterprise Linux 9.8용으로 사전 구축된 Lustre 서버 RPM 패키지를 제공합니다.

Ansible 자동화

NetApp E-Series 스토리지를 사용하는 Lustre는 E-Series 어레이 및 OSS/MDS 서버 노드에 대한 관리 액세스 권한이 있는 제어 노드에서 Ansible 자동화를 사용하여 제공 및 배포됩니다. Ansible 인벤토리는 어레이, 서버, 스토리지 할당, 네트워크 인터페이스 및 HA 클러스터 리소스를 포함하여 원하는 솔루션을 모델링합니다.

NetApp E-Series Lustre Ansible 컬렉션은 SANtricity 및 Host 컬렉션을 사용하여 엔드투엔드 배포를 오케스트레이션합니다. 이 두 컬렉션을 함께 사용하면 E-Series 스토리지 프로비저닝 및 매핑, 서버 운영 체제 준비, NVMe-oF 및 LNet 연결 구성, Lustre 타겟 배포, Pacemaker 리소스 생성 등의 작업을 수행할 수 있습니다. 이러한 자동화를 통해 구성을 반복할 수 있으며 기존 파일 시스템에 구성 요소를 추가하는 작업을 간소화할 수 있습니다.

다음 표는 Ansible 제어 노드에서 사용되는 주요 소프트웨어 패키지를 요약한 것입니다. 현재 패키지 종속성은 "[NetApp E-Series Lustre Ansible 컬렉션](#)"을 참조하십시오.

패키지	의존성 또는 목적
파이썬	Ansible 및 필요한 Python 모듈의 런타임을 제공합니다.
ansible-core	Python이 필요하며 NetApp E-Series Ansible 컬렉션을 실행합니다.
추가 Python 패키지	cryptography, ipaddr, netaddr, passlib, resolvelib, jinja2, 그리고 pyyaml
NetApp E-Series Lustre Ansible 컬렉션	Lustre, LNet, NVMe-oF 호스트 연결 및 Pacemaker 리소스를 구성합니다.
NetApp E-Series SANtricity Ansible 컬렉션	SANtricity 웹 서비스 API를 통해 E-Series 어레이, 스토리지 풀, 볼륨 및 호스트 매핑을 구성합니다.
NetApp E-Series 호스트 Ansible 컬렉션	OSS/MDS 서버 노드를 E-Series 어레이와의 LNet 및 NVMe-oF 연결을 위해 준비합니다.

Lustre 클라이언트 소프트웨어

Lustre 클라이언트 소프트웨어는 시스템이 프론트엔드 LNet 패브릭을 통해 병렬 파일 시스템에 접근할 수 있도록 합니다. 각 클라이언트는 실행 중인 커널 및 서버에 배포된 Lustre 릴리스와 호환되는 Lustre 커널 모듈과 Lustre 클라이언트 유틸리티가 필요합니다.

"[Lustre 지원 매트릭스](#)"을 사용하여 Lustre 버전에 맞는 클라이언트 운영 체제 및 커널을 찾으십시오. NetApp은 사전 빌드된 Lustre 클라이언트 패키지를 제공하지 않습니다. "[netapp-lustre repository](#)"에 있는 NetApp이 제공하는 Lustre 소스 코드에서 클라이언트 운영 체제 및 커널용 클라이언트 패키지를 빌드하십시오. Lustre 클라이언트 패키지가 설치되면 `lustre_client` 역할은 네트워크 인터페이스, LNet을 구성하고 Lustre 파일 시스템에 대한 영구적인 systemd 마운트 유닛을 생성할 수 있습니다.

클라이언트 배포 단계는 "[솔루션 배포](#)"를 참조하십시오.

Lustre with NetApp E-Series 스토리지 - 사이징 가이드

NetApp E-Series 스토리지 빌딩 블록을 사용하여 용량 및 메타데이터 요구 사항에 따라 Lustre의 크기를 조정하고, 용량 추가 시점을 결정하고, 성능에 영향을 미치는 요소를 검토하십시오.

용량 사이징

두 개의 EF80 어레이로 구성된 기본 빌딩 블록은 다음과 같은 Lustre 타겟 레이아웃을 제공합니다. 권장 타겟 배치 및 NVMe-oF 경로에 대한 자세한 내용은 "[기본 구성 요소 볼륨 배포](#)"의 "[하드웨어 구성 요소](#)"를 참조하십시오.

요소	개수	일반적인 볼륨 사이징(대상당)
MGS	1	5~10GiB
MDT	8	크기는 드라이브 용량 및 RAID 레이아웃에 따라 다릅니다.
OST	32	크기는 드라이브 용량 및 RAID 레이아웃에 따라 다릅니다.

각 EF80 어레이는 24개의 NVMe 드라이브를 사용합니다. 지원되는 용량은 기존 볼륨 그룹(MGS/MDT용 RAID 1 및

OST용 RAID 6) 또는 DDP를 사용하는 경우 3.84TB, 7.68TB 및 15.3TB이며, DDP만 사용하는 경우 30.7TB 또는 61.4TB 용량 플래시(QLC) 드라이브도 지원합니다. 1.92TB 드라이브는 현재 이 솔루션에 권장되지 않습니다. 드라이브 슬롯 레이아웃 및 풀 선택에 대한 자세한 내용은 "[하드웨어 구성 요소](#)"를 참조하십시오.

다음 표는 EF80 구성 요소(어레이 2개, 드라이브 48개)의 대략적인 사용 가능 용량을 나타냅니다. 어레이 사용 가능 용량은 최종 Lustre 파일 시스템 사용 가능 용량과 다릅니다. MDT inode 할당, 저널, 과잉 프로비저닝 및 인벤토리 볼륨 비율로 인해 애플리케이션에서 사용할 수 있는 용량이 줄어듭니다.

드라이브 용량	레이아웃(배열별)	대략적인 사용 가능 (구성 요소)
3.84 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	138.08 TB
3.84 TB	DDP(24개 드라이브, 2개 예약됨)	133.63 TB
7.68 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	276.35 TB
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	DDP 전용	1,070.35 TB
61.4 TB	DDP 전용	2,162.70 TB

메타데이터와 데이터의 크기를 별도로 지정한 다음 Ansible 인벤토리에서 볼륨 크기를 설정합니다. 배포 템플릿은 `ldiskfs`를 사용하여 대상을 포맷하고, MDT는 Lustre의 기본 inode 비율을 사용하며, OST 템플릿은 ``-i 4096``을 설정합니다.

- **메타데이터(MDT):** 예상 파일 수를 기준으로 MDT 크기를 조정하십시오. 파일 수 증가를 고려하여 예상 파일 수의 약 두 배를 계획하십시오. MDT 크기가 부족하면 OST 용량이 남아 있더라도 새 파일 생성이 차단될 수 있습니다.
- **데이터(OST):** 사용 가능한 용량과 처리량 요구 사항을 기준으로 OST 크기를 지정합니다.
- **MGS:** 파일 시스템 구성에만 소용량 관리 타겟(5~10GiB)을 사용하십시오.

선택한 드라이브 용량에 맞게 인벤토리에서 MDT 및 OST 볼륨 크기를 조정하십시오.

`format_options.mkfsoptions``을 ``eseries_lustre_filesystem_mdt`` 또는 ``eseries_lustre_filesystem_ost``에서 변경하는 것은 사이트에서 다른 inode 비율이 필요한 경우에만 수행하십시오. inode 비율에 대한 배경 정보는 "[Lustre Wiki: Lustre 튜닝](#)"을 참조하십시오. 빌딩 블록을 추가해야 하는 시점은 **[확장]**을 참조하십시오.

확장

용량이나 메타데이터 부하가 필요할 때 구성 요소를 추가합니다.

- **OST 전용:** 파일 수와 메타데이터 로드가 기존 MDT 용량 내에 있지만 데이터 용량 또는 총 처리량이 더 필요한 경우. OST 32개와 OSS/MDS 서버 노드 2개를 추가합니다.
- **MDT+OST:** 파일 수 또는 메타데이터 처리량이 MDT 용량에 근접했거나, 메타데이터 처리량과 데이터 용량 모두 더 필요한 경우. MDT 8개와 OST 32개를 추가합니다.

기존 MDT에서 ``mdt.*.md_stats``을 사용하여 메타데이터가 포화 상태인지 확인하십시오. 각 Pacemaker/Corosync 클러스터를 5개의 빌딩 블록(10개의 OSS/MDS 서버 노드)으로 제한하십시오. 대규모 배포의 경우 빌딩 블록을 여러 HA 클러스터에 균등하게 분산하십시오. 확장 규칙은 "[솔루션 아키텍처](#)"를 참조하십시오.

다음 예는 기본 구성 요소와 OST 전용 구성 요소를 포함하는 파일 시스템을 하나의 HA 클러스터에 나타낸 것입니다.

빌딩 블록	유형	서버	배열	MGS	MDT	OST
BB1	베이스	2	2	1	8	32
BB2	OST 전용	2	2	0	0	32
총		4	4	1	8	64

BB2는 `mgsnode=`을 사용하여 BB1의 MGS에 대해 OST 대상을 등록합니다. BB2의 OST 인덱스는 전역적으로 할당됩니다(예: OST0032~OST0063). 따라서 인덱스가 BB1과 충돌하지 않습니다.

성능

형식적 검증은 Lustre 클라이언트의 IOR, mdtest 및 fio를 사용하여 상대적 처리량, IOPS 및 메타데이터 동작을 특성화하고 HA 페일오버를 검증합니다. 이러한 테스트는 보장된 성능 수치를 제공하지 않습니다. 합성 벤치마크는 최상의 경우 동작을 측정하며 실제 애플리케이션 성능을 반영하지 않을 수 있습니다.

성능은 구성 요소의 수에 비례하여 향상됩니다. 실제 결과는 드라이브 유형 및 풀 레이아웃, NVMe-oF 경로 수, LNet 패브릭 및 클라이언트 수, 데이터 및 메타데이터 I/O의 비율에 따라 달라집니다.

워크로드별 크기 조정 및 성능 권장 사항에 대해서는 NetApp 계정 담당 팀에 문의하십시오.

드라이브 레이아웃 및 볼륨 수에 대해서는 "[하드웨어 구성 요소](#)"를 참조하십시오. 배포 단계에 대해서는 "[솔루션 배포](#)"를 참조하십시오. 현장 수준 인벤토리 지침에 대해서는 "[Lustre Ansible 인벤토리를 사용자 지정합니다](#)"를 참조하십시오.

솔루션 배포

NetApp E-Series 스토리지를 사용하여 Lustre 배포

NetApp E-Series 스토리지를 사용하여 Lustre를 배포하려면 하드웨어, 환경 준비, 인벤토리, Ansible 배포 및 검증 작업을 순서대로 완료하십시오.

필수 조건

배포를 시작하기 전에 다음 사항을 확인하십시오.

- "[사이즈 가이드](#)"의 크기 조정 및 구성 요소 지침을 따르고 "[하드웨어 구성 요소](#)"에 맞는 하드웨어를 선택했습니다.
- 서버, HCA, E-Series, 운영 체제, Lustre, SANtricity OS 및 DOCA-OFED 조합이 "[NetApp 상호 운용성 매트릭스 도구](#)"의 *E-Series Lustre*에 대해 목록에 포함되어 있음을 확인하셨습니다.
- 계획된 빌딩 블록에 필요한 모든 OSS/MDS 서버 노드와 E-Series 어레이가 현장에 도착하여 랙에 설치할 준비가 완료되었습니다.
- 서버 BMC, 어레이 관리 포트 및 Ansible 제어 노드 역할을 할 호스트에 대한 자격 증명과 네트워크 액세스 권한이 있습니다.



NetApp은 "[NetApp E-Series Lustre Ansible 컬렉션](#)"을 사용하여 Lustre HA 클러스터 및 Lustre 클라이언트 배포를 지원합니다. Lustre 타겟, LNet, NVMe-oF 호스트 연결 또는 Pacemaker 리소스를 수동으로 구성하지 마십시오.

배포 워크플로

다음 작업을 순서대로 완료하십시오.

1. "Lustre 하드웨어 랙 및 케이블 연결".

Lustre 서버와 EF80 어레이를 랙에 설치한 다음 백엔드 NVMe-oF 및 프런트엔드 LNet 패브릭을 케이블로 연결합니다.

2. "Lustre 배포 환경 준비".

어레이와 서버를 구성한 다음 제어 노드에 Ansible과 그 종속성을 설치합니다.

3. "Lustre Ansible 인벤토리를 사용자 지정합니다".

사이트 패브릭에 사용할 템플릿을 선택하고 서버, 어레이, 네트워크, 구성 요소 및 자격 증명을 입력합니다.

4. "Lustre HA 클러스터 및 클라이언트 배포".

서버 운영 체제를 준비하고, NVIDIA DOCA-OFED를 설치하고, 메인 배포 플레이북을 실행하고, Lustre 클라이언트를 구성합니다.

5. "Lustre 배포 확인 및 확장".

운영 환경에 사용하기 전에 클러스터 상태, 마운트 및 용량을 확인하십시오. 파일 시스템에 추가 구성 요소가 필요한 경우 기존 인벤토리를 확장하십시오.

관련 정보

- ["NetApp E-Series Lustre Ansible 컬렉션"](#)
- ["NetApp Lustre 소스 저장소"](#)
- ["HA 관리 사용자 가이드"](#)
- ["성능 튜닝 가이드"](#)
- ["Lustre 클라이언트 배포 템플릿"](#)

랙 및 케이블 Lustre 하드웨어

Lustre 서버와 NetApp EF80 어레이를 랙에 설치한 다음, 각 구성 요소에 대해 백엔드 NVMe-oF 및 프런트엔드 LNet 패브릭을 케이블로 연결합니다.

시작하기 전에

HCA 개수, PCIe 슬롯 요구 사항, 경로 요구 사항 및 MTU 지침은 ["하드웨어 구성 요소"](#)에서 검토하십시오.

단계

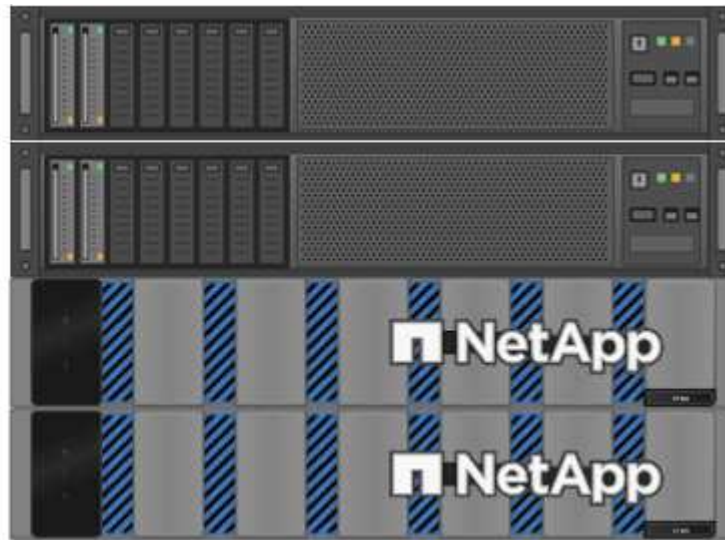
1. 모든 Lustre 서버와 E-Series 스토리지 어레이를 구성 요소 토폴로지에 따라 랙에 장착하고 케이블을 연결하십시오.

Node A

Node B

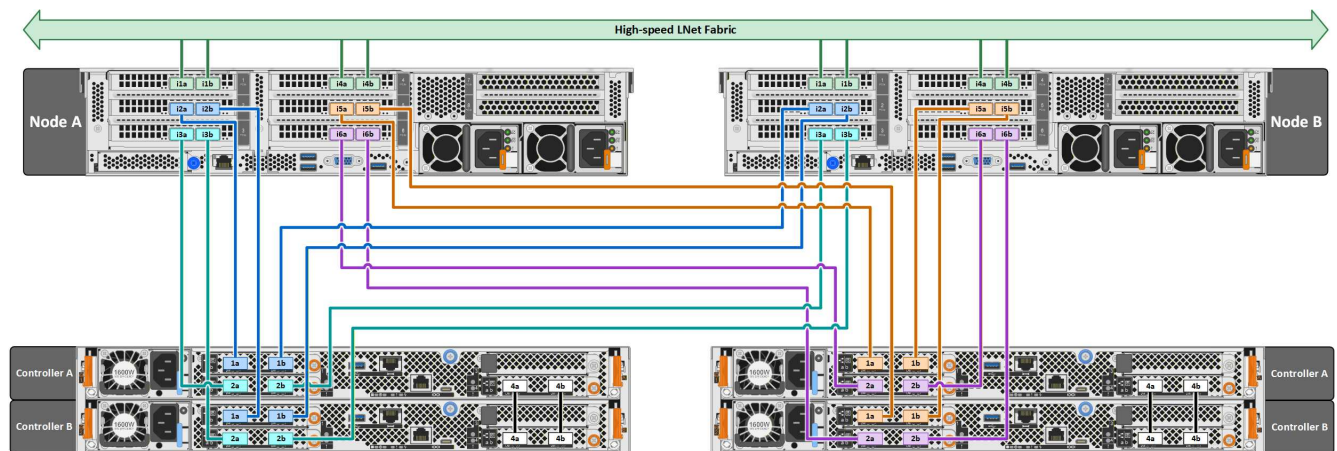
Array 1

Array 2



2. 각 EF80 어레이에 대해 컨트롤러 A의 포트 4a를 컨트롤러 B의 포트 4a에, 그리고 컨트롤러 A의 포트 4b를 컨트롤러 B의 포트 4b에 케이블로 연결하십시오. 이러한 전용 연결은 컨트롤러 간 미러링을 제공하며 호스트 트래픽에는 사용되지 않습니다. 자세한 내용은 "[EF50 및 EF80 컨트롤러 간 미러링 연결 케이블 연결](#)"을 참조하십시오.
3. EF80 6-HCA 토폴로지에 따라 서버와 어레이 간의 백엔드 NVMe-oF 네트워크를 케이블로 연결합니다.

EF80 6-HCA 백엔드 케이블링(RoCE 또는 InfiniBand)



4. 각 Lustre 서버와 Lustre 클라이언트의 프런트엔드 LNet 인터페이스 4개(i1 및 i4 HCA)를 전용 InfiniBand 또는 RoCE 스위치 패브릭에 연결합니다.
5. RoCE를 사용할 때는 우선순위 흐름 제어(PFC)를 통해 무손실 동작이 가능하도록 네트워크 패브릭을 구성하십시오. Ansible 플레이북은 빌딩 블록의 모든 인터페이스에 무손실 RoCE를 구성합니다.

완료한 후

"서버, 어레이 및 Ansible 제어 노드를 준비합니다."

Lustre 배포 환경 준비

각 EF80 어레이와 Lustre 서버에서 관리 액세스 및 기본 설정을 구성한 다음, 솔루션을 배포하기 위한 Ansible 제어 노드를 준비합니다.

E-Series 어레이를 준비합니다.

단계

1. 각 컨트롤러에서 관리 인터페이스를 구성하고 SANtricity System Manager에 연결할 수 있는지 확인하십시오.
2. Ansible 인벤토리에서 사용할 배열 이름과 관리자 자격 증명을 설정합니다.
3. 어레이가 최적의 상태인지, 모든 드라이브가 연결되어 있고 올바르게 보고되는지 확인하십시오.

관리 연결 구성에 대한 지침은 "[E-Series 문서](#)"를 참조하십시오.

Lustre 서버를 준비합니다

단계

1. 각 서버의 베이스보드 관리 컨트롤러(BMC)를 대역 외 액세스용으로 구성하고, UEFI 시스템 설정을 최대 성능으로 설정하십시오.
2. 지원되는 운영 체제(RHEL 또는 Rocky Linux)를 설치한 다음 호스트 이름과 관리 네트워크 포트를 구성하십시오.
3. "[소프트웨어 구성 요소](#)"에 설명된 대로 **E-Series Lustre** 솔루션의 요구 사항을 충족하도록 모든 펌웨어 및 소프트웨어 패키지를 준비하십시오.

Ansible 제어 노드를 준비합니다

모든 Lustre 서버 및 E-Series 어레이에 관리 네트워크 액세스가 가능한 가상 또는 물리적 Linux 시스템을 Ansible 제어 노드로 지정합니다.

다음 단계는 Ubuntu 24.04를 기준으로 설명합니다. 다른 Linux 배포판에 대한 지침은 "[Ansible 설치 문서](#)"를 참조하십시오.

단계

1. Python, pip 및 Python 가상 환경 패키지를 설치합니다.

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

2. Python 가상 환경을 생성하고 활성화합니다.

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

3. 활성화된 가상 환경에 Ansible 및 필요한 Python 패키지를 설치합니다.

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

4. NetApp E-Series Lustre Ansible 컬렉션을 설치합니다. 이 컬렉션을 설치하면 SANtricity 및 Host 컬렉션을 포함한 종속 컬렉션도 함께 설치됩니다.

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

5. 설치된 Ansible, Python 및 컬렉션 버전을 확인하십시오.

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

6. 제어 노드에서 각 Lustre 서버로 암호 없는 SSH 연결을 구성하십시오. ``을 서버의 관리 IP 주소 또는 호스트 이름으로 바꾸십시오.

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

각 Lustre 서버에 대해 이 과정을 반복하십시오. ssh-copy-id



E-Series 어레이에 암호 없는 SSH 연결을 구성하지 마십시오. Ansible은 인벤토리에 정의된 자격 증명을 사용하여 SANtricity 웹 서비스 API를 통해 어레이를 관리합니다.

완료한 후

"Ansible 인벤토리를 사용자 정의합니다".

Lustre Ansible 인벤토리를 사용자 지정합니다

EF80 배포 템플릿의 작업 복사본을 만들고 서버, 어레이, 네트워크 및 Lustre 구성 요소에 맞게 인벤토리를 사용자 지정합니다.

선택한 인벤토리를 해당 패브릭 및 플랫폼 디렉터리 아래에 보관하십시오. 공유 playbooks 디렉터리와 passwords.yml 파일을 `building\_blocks`의 루트에 유지하십시오.

배포 템플릿을 선택합니다

사이트의 네트워크 패브릭과 일치하는 템플릿에서 시작합니다:

- **InfiniBand:** "IB_H2_IB_DAC_A2_EF80/lustre_building_block_cluster_1"
- **RoCE:** "ROCE_H2_ROCE_DAC_A2_EF80/lustre_building_block_cluster_1"

사이트별 설정 구성

인벤토리에서 사이트별 값을 설정하고 로컬 저장소 및 udev 규칙 설정을 함께 지정하십시오. 표의 줄 참조는 ansible-lustre release/1.0.0 태그에 있는 RoCE EF80 템플릿 파일을 가리킵니다. InfiniBand 템플릿은 LNet 인터페이스 변수를 제외하고 동일한 변수 이름과 구조를 사용하며, 해당 변수는 표에 명시되어 있습니다.

필드	Ansible 변수	템플릿 파일 (release/1.0.0)	범위	참고
파일 시스템 이름	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"	클러스터 전체	format_options 아래에 중첩됩니다(MGT, MDT, OST 맵의 경우). 최대 8자입니다. 세 위치 모두에 동일한 값을 사용하십시오. 최상위 fsname 키는 아무런 효과가 없습니다.
표적 등록을 위한 MGS NID	eseries_lustre_filesystem_mdt. format_options. mgsnode, eseries_lustre_filesystem_ost. format_options. mgsnode	"ha_cluster.yml#L98", "#L118"	클러스터 전체	MDT 및 OST 아래에 중첩됩니다 format_options. HA 쌍의 두 노드 모두에서 프론트엔드 LNet NID를 포함합니다. 기본 MGS 소유자로 구성된 노드의 NID를 먼저 나열하고 그 뒤에 파트너 노드의 NID를 나열합니다.
Pacemaker 클러스터 이름	eseries_lustre_pacemaker_cib_properties_overrides.cluster_properties.cluster-name	"ha_cluster.yml#L228"	클러스터 전체	Pacemaker CIB 속성 재정의 아래에 중첩됩니다. HA 클러스터에 대한 고유 이름입니다.
BMC 펜싱 주소	eseries_lustre_pacemaker_fencing_agents.fence_redfish.ip	"ha_cluster.yml#L211", "#L214", "#L217", "#L220"	Lustre 서버당 반복	각 서버 노드의 Redfish BMC IP 주소입니다.
알림 이메일 수신자	eseries_lustre_alert_email_list	"ha_cluster.yml#L233"	클러스터 전체	HA 리소스 및 장애 알림 수신자입니다.
메일 도메인	eseries_lustre_mail_service_overrides.conf.mydomain	"ha_cluster.yml#L240"	클러스터 전체	Postfix 도메인은 알림 이메일을 보내는 데 사용됩니다.
NTP 시간 소스	eseries_lustre_time_sync_overrides.server_pools	"ha_cluster.yml#L250"	클러스터 전체	클러스터 노드의 시간 소스입니다. NTP 풀 또는 하나 이상의 개별 NTP 서버를 지정합니다. 필수 `pool` 또는 `server` 지시문과 `iburst` 등의 옵션을 포함합니다.

필드	Ansible 변수	템플릿 파일 (release/1.0.0)	범위	참고
인터페이스 PCI-to-name udev 맵	eseries_ip_udev _rules	"ha_cluster.yml#L172"	HCA 슬롯 레이아웃당 반복	각 PCI 슬롯을 영구 인터페이스 이름(L182의 값)에 매핑합니다. `lspci -nnvmm`을 사용하여 수집합니다.
로컬 패키지 저장소(선택 사항)	eseries_common_ yum_repos	"ha_cluster.yml#L256", "#L261"	클러스터 전체	에어갭 설치의 경우 해당 블록의 주석 처리를 해제하고 저장소 url(L264)를 설정하십시오.
서버 관리 IP	ansible_host	"host_vars/lustre_01.yml#L1 0"	Lustre 서버당 반복	서버당 파일 하나. host_vars/lustre_NN.y ml
NVMe-oF 백엔드 인터페이스	eseries_nvme_ro ce_interfaces (RoCE) / eseries_nvme_ib _interfaces (InfiniBand)	"RoCE host_vars/lustre_01.yml#L2 1", "InfiniBand host_vars/lustre_01.yml#L1 7"	Lustre 서버당 반복	E-Series 어레이로의 NVMe- oF 스토리지 트래픽에 사용되는 백엔드 인터페이스 주소입니다.
LNet 클러스터 인터페이스	eseries_roce_in terfaces (RoCE) / eseries_ipoib_i nterfaces (InfiniBand)	"RoCE host_vars/lustre_01.yml#L4 7", "InfiniBand host_vars/lustre_01.yml#L3 8"	Lustre 서버당 반복	프론트엔드 LNet 인터페이스 주소(L56의 RoCE 값, InfiniBand L47의 InfiniBand 값).
Corosync 노드 IP	eseries_lustre_ corosync_node_i ps	"host_vars/lustre_01.yml#L5 8"	Lustre 서버당 반복	HA 클러스터의 모든 노드 IP 주소를 나열합니다(L62 레벨 값).
Lustre 타겟 서비스 NID	lustre_target_s ervice_nodes	"host_vars/lustre_01.yml#L6 4"	Lustre 서버당 반복	HA 쌍의 두 노드 모두에서 프론트엔드 LNet NID(@o2ib)를 포함하여 장애 조치 후에도 대상 서비스에 계속 접근할 수 있도록 합니다(L67의 값 참조).
어레이 관리 IP	ansible_host	"host_vars/netapp_01.yml# L13"	E-Series 어레이별로 반복합니다.	컨트롤러 하나의 관리 IP 주소입니다. API URL은 이 주소에서 파생됩니다.

필드	Ansible 변수	템플릿 파일 (release/1.0.0)	범위	참고
구성 요소 그룹 멤버십	mgs / mds_NN / oss_NN	"lustre_inventory.yml"	빌딩 블록당	각 인벤토리 호스트 이름은 해당 host_vars 파일의 기본 이름과 일치해야 합니다. 예를 들어, 호스트 `lustre_01`는 `host_vars/lustre_01.yml`를 사용하고, `netapp_01`는 `host_vars/netapp_01.yml`를 사용합니다. MGS 그룹은 첫 번째 빌딩 블록에만 포함하십시오. MDS 그룹은 기본 빌딩 블록과 각 MDT+OST 확장 빌딩 블록에 포함하고, OST 전용 확장 빌딩 블록에서는 MDS 그룹을 제외하십시오. `mgsgnode`를 사용하여 기존 MGS에 대한 확장 대상을 등록하십시오.

각 호스트 및 빌딩 블록에 대해 설정을 반복합니다. Lustre 서버당 하나의 host_vars/lustre_NN.yml 파일과 어레이당 하나의 host_vars/netapp_NN.yml 파일을 생성하고, 추가 빌딩 블록마다 인벤토리 그룹을 반복합니다.

배포 플레이북은 공유 building_blocks/passwords.yml 파일에서 배열, Pacemaker 및 BMC 자격 증명을 로드합니다. 배포 전에 이 파일을 채우고 Ansible Vault로 암호화하거나 `--extra-vars`를 사용하여 런타임에 안전하게 값을 재정의하십시오. 평문 자격 증명을 소스 제어 시스템에 저장하지 마십시오.



배포 템플릿의 IP 주소, 호스트 이름, 인터페이스 이름 및 볼륨 사이징 등이 예시입니다. 플레이북을 실행하기 전에 환경에 대한 모든 인벤토리 파일을 수정하십시오.

완료한 후

"Lustre HA 클러스터 및 클라이언트 배포".

Lustre HA 클러스터 및 클라이언트 배포

서버 운영 체제를 준비하고, NVIDIA DOCA-OFED를 설치하고, Lustre HA 클러스터를 배포하고, NetApp Ansible 플레이북을 사용하여 Lustre 클라이언트를 구성합니다.

시작하기 전에, "Lustre Ansible 인벤토리를 사용자 지정합니다"에 설명된 목록을 작성하고 안전하게 보관하십시오.

Lustre HA 클러스터 배포

단계

1. 작업 복사본의 building_blocks/playbooks 디렉터리에서 Lustre 서버의 운영 체제를 준비합니다. ``를 사용자 지정 인벤토리 디렉터리의 경로로 바꿉니다.

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

```
deploy_lustre_os/deploy_lustre_os_playbook.yml
```

2. 각 Lustre 서버에 NVIDIA DOCA-OFED를 설치합니다. 이전 단계에서 설치한 커널을 기반으로 커널 모듈을 빌드합니다. 사용 중인 커널에 맞는 절차를 "[NVIDIA DOCA-Host 설치 및 업그레이드](#)"에서 따르십시오. 다음 예는 RHEL 또는 Rocky Linux에 DOCA-OFED를 설치하는 방법을 보여줍니다.
 - a. DOCA 호스트 저장소 패키지를 설치하고 패키지 캐시를 새로 고칩니다. ``을 운영 체제 및 아키텍처에 맞는 DOCA 호스트 패키지로 교체합니다.

```
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. 실행 중인 커널용 DOCA 커널 모듈을 빌드합니다. 스크립트는 생성된 패키지의 경로를 출력합니다.

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

- c. 스크립트가 생성한 커널 모듈 저장소 패키지를 설치한 다음 패키지 캐시를 새로 고칩니다. ``을 이전 단계의 경로로 바꾸십시오.

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

- d. DOCA-OFED 사용자 공간, 커널 및 NVMe 패키지를 설치합니다. ``을(를) 실행 중인 커널 버전으로 교체합니다.

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

3. 메인 배포 플레이북을 실행합니다.

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



배포 규모에 따라 Ansible이 병렬로 구성할 호스트 수를 제어하려면 `--forks`을 조정합니다. 예를 들어, 더 큰 클러스터의 경우 `--forks 20`를 추가합니다.

플레이북은 E-Series 볼륨 그룹 또는 DDP 풀 및 볼륨을 프로비저닝하고, NVMe-oF 호스트 연결을 구성하고, Lustre

타겟을 포맷 및 마운트하고, LNet을 구성하고, 성능 튜닝을 적용하고, Pacemaker HA 리소스를 구성합니다.



하나의 구성 요소로 이루어진 배포는 2노드 Pacemaker 클러스터를 구성합니다. 노드 A는 장애 조치 시 추가 투표권을 갖습니다. 2노드 클러스터에서 쿼럼을 달성하려면 qdevice 구성을 고려하십시오. 배포에 여러 구성 요소가 포함된 경우, 각 구성 요소는 문서에 명시된 클러스터 제한까지 더 큰 Pacemaker 클러스터에 2노드 서버 쌍을 제공합니다. "HA 관리 사용자 가이드"의 펜싱 및 쿼럼 구성을 검토하여 노드 장애 발생 시 클러스터가 안전하게 타겟을 복구할 수 있도록 하십시오.

Lustre 클라이언트 배포

클라이언트 배포 템플릿을 사용자 지정하고 클라이언트 플레이북을 실행하여 파일 시스템 액세스가 필요한 모든 시스템에 Lustre 클라이언트를 배포할 수 있습니다.

단계

- 호환되는 클라이언트 운영 체제와 커널을 "Lustre 지원 매트릭스"에서 선택하십시오. 해당 커널용 Lustre 클라이언트 패키지가 아직 설치되지 않은 경우 "netapp-lustre"에서 빌드 및 설치하십시오.
- 공유 passwords.yml 파일을 포함하여 clients 템플릿 디렉토리의 작업 복사본을 생성하고, 패브릭과 일치하는 템플릿을 선택합니다.
 - **InfiniBand:** "IB_lustre_cluster_clients"
 - **RoCE:** "ROCE_lustre_cluster_clients"
- 클라이언트에 맞게 lustre_client_inventory.yml, host_vars, 및 group_vars`를 사용자 지정하세요. 표의 줄 참조는 `ansible-lustre release/1.0.0 태그에 있는 RoCE 클라이언트 템플릿 파일을 가리킵니다. InfiniBand 클라이언트 템플릿은 별도로 명시된 경우를 제외하고 동일한 변수를 사용합니다.

필드	Ansible 변수	템플릿 파일 (release/1.0.0)	범위	참고
클라이언트 호스트	lustre_clients	"lustre_client_inventory.yml #L4"	고객별	각 클라이언트 호스트 이름을 나열합니다.
클라이언트 SSH 사용자 및 비밀번호 되기	ssh_client_user / ssh_client_become_pass	"passwords.yml#L2"	클러스터 전체	SSH 사용자가 아닌 경우에만 become 비밀번호를 설정하십시오 root.
LNet 인터페이스	eseries_lustre_lnet	"group_vars/all.yml#L7"	클러스터 전체	클라이언트 측 LNet 인터페이스 이름(L13의 값).
마운트 MGS NID	lustre_client_mounts.mgsnode	"group_vars/all.yml#L17"	클러스터 전체	파일 시스템을 마운트하는 데 사용되는 MGS NID(L20의 값).
파일 시스템 이름	lustre_client_mounts.fsname	"group_vars/all.yml#L21"	클러스터 전체	클러스터와 일치해야 합니다 fsname.
마운트 포인트	lustre_client_mounts.mount_point	"group_vars/all.yml#L22"	클러스터 전체	클라이언트 마운트 디렉터리.
클라이언트 관리 IP	ansible_host	"host_vars/client_01.yml#L4"	고객별로 반복	One host_vars/client_NN. yaml 클라이언트당 파일 하나.

필드	Ansible 변수	템플릿 파일 (release/1.0.0)	범위	참고
스토리지 패브릭 인터페이스	eseries_roce_in- terfaces	"host_vars/client_01.yml#L 10"	고객별로 반복	프론트엔드 인터페이스 주소(L15 값). InfiniBand 템플릿에서 'eseries_ipoib_interfaces' 을(를) 사용합니다.

각 클라이언트에 대해 설정을 반복합니다. 클라이언트별로 `host_vars/client_NN.yml` 파일을 하나씩 생성하고 해당 호스트를 `lustre_client_inventory.yml`에 추가합니다. 공유 `clients/passwords.yml` 파일에 데이터를 입력하고 클라이언트 플레이북을 실행하기 전에 Ansible Vault로 암호화합니다.

4. 클라이언트 템플릿 디렉터리에서 클라이언트 플레이북을 실행합니다.

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

클라이언트 플레이북은 클라이언트 네트워크 인터페이스와 LNet을 구성하고, Lustre 파일 시스템에 대한 영구적인 systemd 마운트 유닛을 생성할 수 있습니다.

완료한 후

"배포를 확인합니다".

Lustre 배포 확인 및 확장

운영 환경에 사용하기 전에 클러스터 상태, Lustre 마운트 및 파일 시스템 용량을 확인한 다음, 필요할 때 동일한 인벤토리를 사용하여 구성 요소를 추가하십시오.

배포를 확인합니다

단계

1. Lustre 서버에서 모든 Pacemaker 리소스가 예상 노드에서 시작되었는지 확인하십시오.

```
pcs status
```

2. Lustre 클라이언트에서 Lustre 파일 시스템이 마운트되었는지 확인하십시오.

```
mount -t lustre
```

3. Lustre 클라이언트에서 파일 시스템 용량과 MDT 및 OST 대상이 표시되는지 확인하십시오.

```
lfs df -h
```

IOR, mdtest 및 fio를 사용한 성능 검증에 대해서는 ["사이즈 가이드"](#)을 참조하십시오.

제어된 대상 마이그레이션 또는 HA 페일오버 테스트를 위해서는 ["HA 관리 사용자 가이드"](#)에 있는 절차를 따르십시오.

파일 시스템을 확장합니다

용량 또는 메타데이터 성능을 추가하려면 기존 인벤토리를 추가 빌딩 블록으로 확장하십시오. 새 빌딩 블록에 대해 별도의 인벤토리를 생성하지 마십시오. 플레이북은 자동으로 고유한 MDT 및 OST 인덱스를 할당하고 기존 MGS에 새 타겟을 등록합니다.

단계

1. 새로운 빌딩 블록 호스트, 어레이 및 리소스 그룹(MDT+OST 또는 OST 전용)을 원래 배포에 사용했던 인벤토리 및 `host_vars` 및 `group_vars` 파일에 추가합니다.
2. 업데이트된 인벤토리를 사용하여 메인 배포 플레이북을 다시 실행하십시오.

확장 규칙은 ["솔루션 아키텍처"](#)을, 각 구성 요소 유형을 추가해야 하는 시점에 대한 지침은 ["사이즈 가이드"](#)을 참조하십시오.

Lustre with NetApp E-Series 스토리지 - 관련 자료

Lustre with NetApp E-Series 스토리지 배포를 계획하거나 확장할 때 관련 문서, 툴 및 소프트웨어는 다음 링크를 참조하십시오. 사이징 및 배포 단계는 ["사이즈 가이드"](#) 및 ["솔루션 배포"](#)을 참조하십시오.

NetApp 리소스

- ["NetApp 상호 운용성 매트릭스 도구"](#). *E-Series Lustre*를 검색합니다.
- ["NetApp EF 시리즈 올플래시 어레이 데이터시트"](#)
- ["NetApp SANtricity 시스템 관리자 설명서"](#)
- ["NetApp ansible-lustre 컬렉션"](#). 재고 관리 현장 지침은 ["Lustre Ansible 인벤토리를 사용자 지정합니다"](#)을 참조하십시오.
- ["NetApp Lustre 소스 저장소"](#). NetApp은 Rocky Linux 9.8 및 Red Hat Enterprise Linux 9.8용으로 미리 빌드된 Lustre 서버 RPM 패키지를 제공합니다. NetApp 소스에서 클라이언트 운영 체제 및 커널에 맞는 Lustre 클라이언트 패키지를 빌드하십시오.

Lustre 커뮤니티

- ["Lustre 파일 시스템"](#)
- ["Lustre 지원 매트릭스"](#) 클라이언트 운영 체제 및 커널의 경우
- ["Lustre Wiki: Lustre 튜닝"](#)

관련 **NetApp AI** 인프라 솔루션

- ["E-Series 스토리지를 갖춘 NetApp 의 BeeGFS"](#)
- ["NetApp E-Series 스토리지를 사용한 IBM Spectrum Scale 배포"](#)
- ["NetApp EF 시리즈를 탑재한 NVIDIA DGX SuperPOD"](#)

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.