Lustre mit NetApp E-Series Speicher

NetApp artificial intelligence solutions

NetApp
August 31, 2026

This PDF was generated from https://docs.netapp.com/de-de/netapp-solutions-ai/infra/ai-eseries-lustre-intro.html on August 31, 2026. Always check docs.netapp.com for the latest.

Inhalt

Lustre mit NetApp E-Series Speicher

Lustre mit NetApp E-Series Storage – Einführung

Lustre mit NetApp E-Series Storage ist eine validierte parallele Dateisystemlösung für KI- und HPC-Workloads, die aus wiederverwendbaren Bausteinen von HA-Serverpaaren und E-Series All-Flash-Arrays besteht.

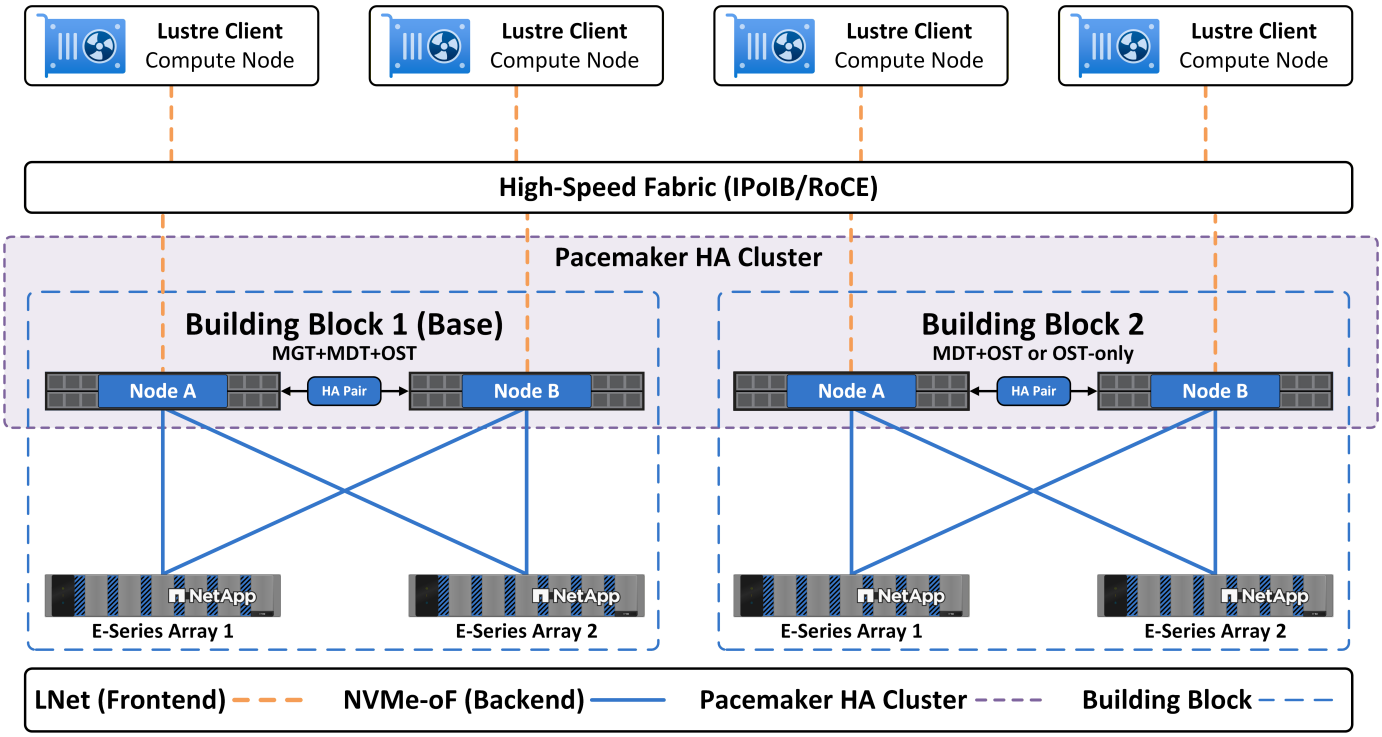
Lösungsübersicht

Lustre mit NetApp E-Series Storage kombiniert das Open-Source-Paralleldateisystem Lustre mit NetApp E-Series Blockspeicher und bietet eine skalierbare, leistungsstarke Grundlage für datenintensive Workloads. Jeder Baustein kombiniert zwei für Hochverfügbarkeit (HA) konfigurierte OSS/MDS-Serverknoten mit zwei E-Series All-Flash-Arrays. Die Backend-NVMe-oF-Konnektivität überträgt die Block-I/O an die Arrays, und ein separates LNet-Fabric verbindet Clients und OSS/MDS-Serverknoten für den parallelen Zugriff auf das Dateisystem.

Ein Baustein beherbergt den Management-Server (MGS) sowie die initialen Metadatenziele (MDTs) und Objektspeicherziele (OSTs). Zusätzliche Bausteine registrieren sich beim Basis-MGS, um die Metadatenkapazität, die Datenkapazität und den Gesamtdurchsatz mit steigenden Arbeitslastanforderungen zu skalieren.

Die folgende Abbildung zeigt ein Beispiel für eine Bereitstellung mit mehreren Bausteinen und Lustre Clients, die über LNet verbunden sind.

Lustre mit NetApp E-Series Speicherlösung Übersicht



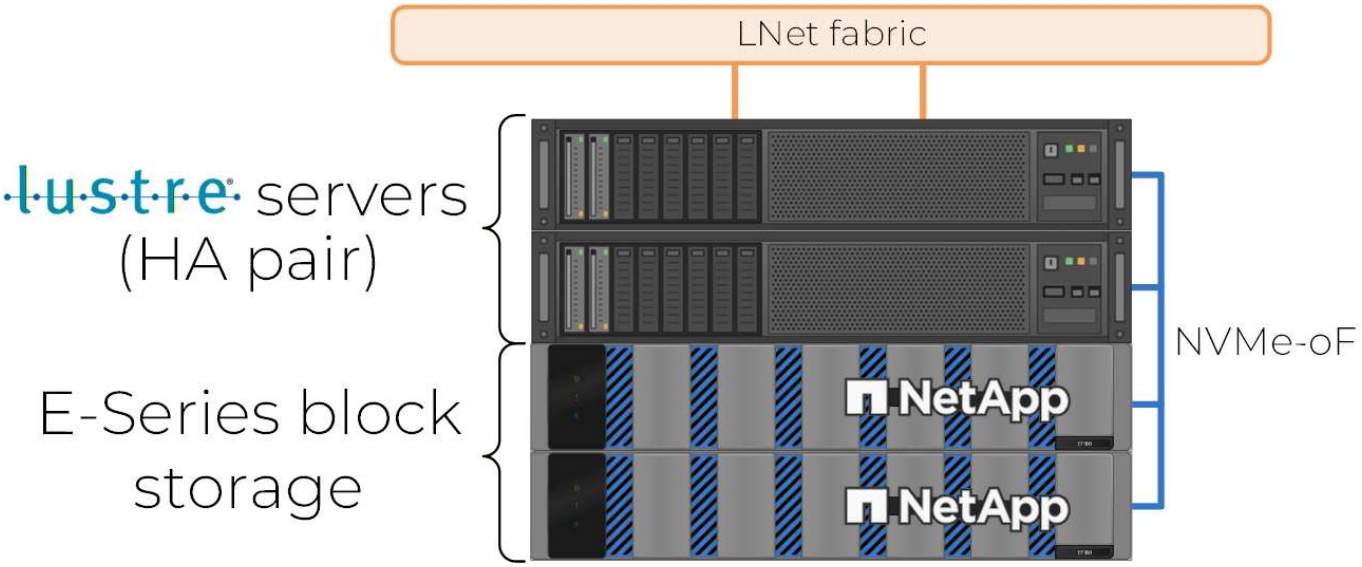
Lustre mit NetApp E-Series Storage - Lösungsarchitektur

Es wird erläutert, wie Lustre mit NetApp E-Series Storage Bausteine, Hochverfügbarkeit, Netzwerktopologie und Clients nutzt, sodass Kapazität, Leistung und Failover geplant werden können.

Baustein-Design

Ein Baustein ist die grundlegende skalierbare Einheit der NetApp E-Series Lustre Lösung. Jeder Baustein besteht aus zwei Server-Nodes, die als Hochverfügbarkeits-(HA)-Paar konfiguriert sind und Lustre Object Storage Server (OSS)- und Metadata Server (MDS)-Funktionen bereitstellen, zwei NetApp E-Series All-Flash-Arrays, dedizierter Backend-NVMe over Fabrics (NVMe-oF)-Konnektivität zwischen Servern und Arrays sowie dedizierter Frontend-Lustre-Netzwerk-(LNet)-Konnektivität für Clients und die Inter-Node-Kommunikation. Ein Pacemaker- und Corosync-HA-Cluster kann das Server-Paar aus einem oder mehreren Bausteinen enthalten. Die NetApp `ansible-lustre` Collection nutzt Ansible-Automatisierung zur Bereitstellung und Konfiguration der Lustre-Services.

Lustre Baustein



Die Bausteine skalieren entsprechend den Anforderungen an das Wachstum des Dateisystems. Jedes Dateisystem benötigt einen Basisbaustein. Der Basisbaustein hostet den Management-Server (MGS) auf einem Management-Ziel (MGT) und stellt anfängliche Kapazität für Metadaten-Ziel (MDT) und Objektspeicher-Ziel (OST) bereit. Zusätzliche Bausteine erweitern das Dateisystem und registrieren ihre MDT- und OST-Ziele beim MGS des Basisbausteins. Der modulare Ansatz ermöglicht eine unabhängige Skalierung von Kapazität und Leistung entsprechend den Anforderungen der Arbeitslast.

NetApp empfiehlt für die meisten Bereitstellungen die folgenden Baustein Konfigurationsprofile. Die angegebenen Anzahlen von MGS, MDT und OST sind empfohlene Standardwerte, die für maximale Leistung und Kapazität der E-Series Speicherarrays optimiert sind. Die Anzahl der Ziel-Volumes, die Volumengrößen und die Zuordnung der E-Series Laufwerke im Ansible-Inventar können an die Anforderungen der Workloads angepasst werden. So können beispielsweise Bereitstellungen mit höherem Bedarf an Metadatenspeicherkapazität mehr MDTs und weniger OSTs innerhalb derselben Baustein Hardware vorsehen.

Die folgende Tabelle fasst Bausteintypen und die empfohlenen Zielanzahlen zusammen.

Typ	MGS	MDTs	OSTs	Verwenden
Basis	1	8	32	Erster Baustein eines Dateisystems. Enthält die MGS- sowie die anfängliche MDT- und OST-Kapazität.
MDT+OST	0	8	32	Fügt Metadaten und Datenkapazität hinzu. Registrierung gegen den Basis-Baustein MGS.
nur OST	0	0	32	Fügt lediglich Datenkapazität hinzu. Registrierung gegen den Basis-Baustein MGS.

- **Laufwerkstyp und Pool-Layout:** E-Series unterstützt herkömmliche RAID-Volume-Gruppen und Dynamic Disk Pools (DDP). Für TLC NVMe-Laufwerke werden RAID 6-Volume-Gruppen für OST-Speicher und RAID 1-Volume-Gruppen für MGS- und MDT-Speicher verwendet. Für QLC NVMe-Laufwerke wird DDP für den gemeinsam genutzten Laufwerkspool verwendet.
- **Zielverteilung:** Die Lustre-Ziele in jedem Baustein sind gleichmäßig auf beide OSS/MDS-Serverknoten und beide E-Series-Arrays verteilt. Das Layout verteilt die E/A auf die Server-CPUs und Array-Controller, um die NVMe-oF-Pfadleistung zu verbessern und Redundanz zu gewährleisten. Siehe "Zielverteilung und NVMe-oF Konnektivität" in "Hardwarekomponenten".

Unterstützte Betriebssysteme, Lustre-Versionen, Array-Firmware und zugehörige Baustein-Komponenten sind in "NetApp Interoperabilitätsmatrix Tool (IMT)" unter **E-Series Lustre** aufgeführt. Detaillierte Angaben zu Volume-Anzahlen, Laufwerkslayouts und Dimensionierungsrichtlinien finden sich in "Hardwarekomponenten". Für Softwarekomponenten siehe "Softwarekomponenten".

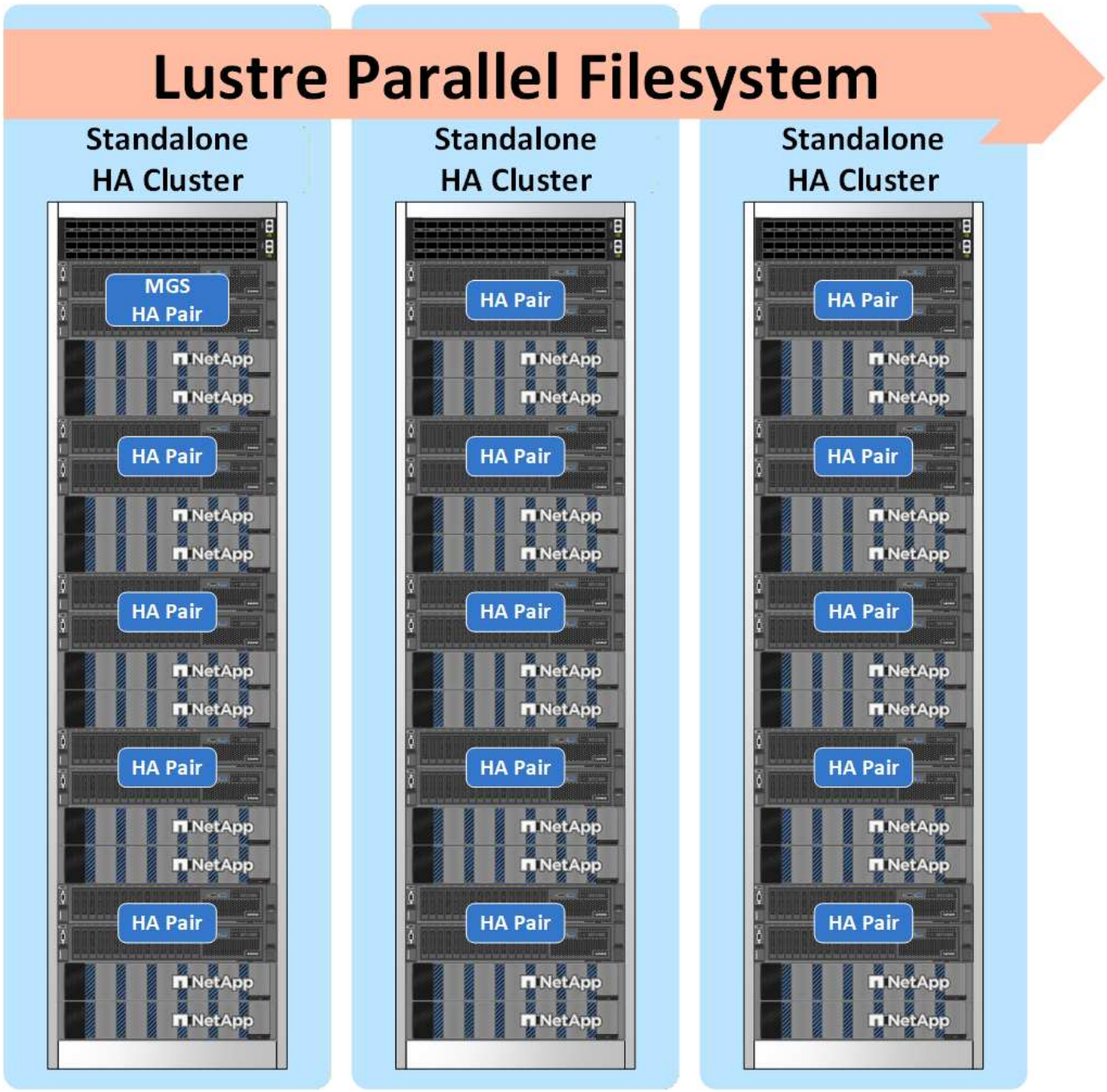
Skalierungsempfehlungen

Das Lustre Dateisystem skaliert durch Hinzufügen von Bausteinen, wenn die Metadatenkapazität, die Datenkapazität oder beides an ihre Grenzen stößt. MDTs werden basierend auf der Dateianzahl und den Metadaten-IOPS skaliert, OSTs basierend auf der Kapazität und den Anforderungen an den Durchsatz.

1. **Ein Baustein pro Dateisystem:** Genau ein Baustein wird bereitgestellt; er hostet die MGS sowie die anfänglichen MDT- und OST-Ziele.
2. **Zusätzliches Bausteinprofil:** Ein reiner OST-Baustein wird hinzugefügt, wenn die vorhandene MDT-Kapazität ausreicht und die Datenkapazität oder die Anforderungen an den Durchsatz steigen müssen. Ein MDT+OST-Baustein wird hinzugefügt, wenn die Metadatenleistung oder die Namespace-Kapazität zum Engpass wird.
3. **HA-Cluster-Limit:** Jeder Pacemaker und Corosync HA-Cluster ist auf fünf Bausteine (zehn Lustre-Serverknoten) zu begrenzen. Größere Bereitstellungen sind gleichmäßig auf mehrere HA-Cluster aufzuteilen, um Ressourcenengpässe in großen Konfigurationen zu vermeiden.

Die folgende Abbildung zeigt bis zu fünf Bausteine, die in einem 42U-Rack gestapelt sind (ein Pacemaker HA-Cluster pro Rack). Mehrere Racks können an einem einzigen Lustre-Dateisystem teilnehmen; nur der Basis-Baustein hostet das MGS.

Lustre paralleles Dateisystem Rack-Skalierung



Beispiele zur Dimensionierung sind unter "Größenberatung" zu finden.

HA Architektur

Lustre mit NetApp E-Series Storage verwendet eine gemeinsam genutzte Hochverfügbarkeitsarchitektur (HA) mit gemeinsam genutzten Festplatten, die in NetApp E-Series Storage integriert ist. Pacemaker verwaltet die HA-Ressourcen, und Corosync stellt die Clusterzugehörigkeit und das Messaging bereit. Gemeinsam verwalten sie das Lustre-Ziel-Failover zwischen den beiden OSS/MDS-Serverknoten in jedem Baustein. Multipath NVMe-oF verbindet beide OSS/MDS-Serverknoten mit denselben E-Series Volumes, sodass jeder Knoten bei Bedarf die Zieldienste übernehmen kann.

Jedes MGS, MDT und OST ist als HA-Ressource mit seinen Abhängigkeiten in einer Pacemaker-

Ressourcengruppe konfiguriert. Pacemaker stellt sicher, dass Ressourcen in der richtigen Reihenfolge gestartet und gestoppt werden, auf demselben Node verbleiben und jeweils nur auf einem Node ausgeführt werden. Gleichzeitiger Zugriff auf dasselbe Ziel-Volume von beiden Nodes aus könnte das zugrunde liegende Dateisystem beschädigen.

- **Target-Failover:** Beide OSS/MDS-Serverknoten sind gleichberechtigte Partner im Cluster, jedoch ist jedes Lustre-Ziel jeweils nur auf einem Knoten aktiv. Eine Pacemaker-Überwachungsressource überwacht den Zustand jedes Ziels und seiner Abhängigkeiten und löst ein Failover aus, sobald ein Ziel auf seinem aktuellen Knoten nicht mehr verfügbar ist. Im Fehlerfall startet Pacemaker die betroffene Ressourcengruppe auf dem Partnerknoten in der korrekten Reihenfolge neu, und Clients verbinden sich transparent mit dem Ziel an seinem neuen Standort, sobald die Dienste wieder verfügbar sind. Da jedes Ziel nur auf einem einzigen Knoten aktiviert ist, wird das Dateisystem niemals gleichzeitigen Schreibvorgängen von beiden Knoten ausgesetzt.
- **Gemeinsamer Speicherzugriff:** Multipath NVMe-oF-Pfade verbinden jeden OSS/MDS-Serverknoten mit allen E-Series Controllern im Baustein, sodass jedes Ziel-Volume von jedem Knoten aus erreichbar ist. Wenn ein Serverknoten ausfällt, verfügt der Partnerknoten bereits über aktive Pfade zu denselben Volumes und kann die Ziele übernehmen, ohne dass die Speicherkonnektivität neu konfiguriert werden muss. Fällt ein Array-Controller oder ein Pfad aus, leitet Multipath die I/O an einen verbleibenden Controller um. Diese doppelte Redundanz ermöglicht es Lustre-Zielen, zwischen OSS/MDS-Serverknoten oder Array-Controllern zu wechseln, ohne den Zugriff auf die zugrunde liegenden Volumes zu verlieren.
- **STONITH-Fencing:** Im Fehlerfall kann Pacemaker manchmal nicht mit dem fehlerhaften Knoten kommunizieren, um zu bestätigen, dass die Ziele gestoppt sind. Bevor diese Ziele an anderer Stelle neu gestartet werden, isoliert Pacemaker den fehlerhaften Knoten, idealerweise durch Trennen der Stromversorgung, um sicherzustellen, dass er abgeschaltet ist. Fencing verhindert ein Split-Brain-Szenario, bei dem beide Knoten gleichzeitig auf dasselbe Ziel zugreifen und das zugrunde liegende Dateisystem beschädigen, wodurch die Datenintegrität während des Failovers erhalten bleibt. NetApp empfiehlt `fence_redfish` für Server mit Redfish-fähigen Baseboard Management Controllern (BMCs). Andere Fencing-Agenten, wie `fence_apc`, werden ebenfalls unterstützt.

Weitere Informationen zur Clusterverwaltung und Fencing-Konfiguration enthält der "HA Benutzerhandbuch" im "ansible-lustre Collection".

Netzwerkarchitektur

Die Lösung nutzt separate Netzwerkpfade für den Datenverkehr des Storage-Backends und den LNet-Frontend-Datenverkehr.

- **Backend (NVMe-oF):** OSS/MDS-Serverknoten verbinden sich über NVMe-oF mit E-Series Arrays mittels NVMe/InfiniBand oder NVMe/RoCE. Der NVMe-oF-Pfad überträgt Block-I/O zwischen Lustre-Zieldiensten und E-Series Volumes. Für EF80-Sechs-HCA-Backend-Verkabelung siehe "Rack- und Kabel-Lustre-Hardware". Für bevorzugte Zielplatzierung über Knoten und Arrays siehe "Zielverteilung" in "Hardwarekomponenten".
- **Frontend (LNet):** Lustre-Clients und OSS/MDS-Serverknoten kommunizieren über LNet mit dem `@o2ib` Netzwerktyp und dem NVIDIA OFED-Stack. InfiniBand und RoCE sind beides validierte Frontend-Transporte. Die ansible-lustre Collection konfiguriert alle Frontend-Schnittstellen für Multi-Rail-LNet, das mehrere Ports aggregiert, um die Bandbreite zu erhöhen und Pfadredundanz für Client- und Knotenverkehr bereitzustellen.
- **Management und Cluster:** Ein Out-of-Band-Managementnetzwerk ermöglicht Servermanagement, BMC-Zugriff und Array-Management. STONITH-Fencing-Agenten, wie `fence_redfish`, nutzen dieses Netzwerk, um die Server-BMCs zu erreichen und die Stromversorgung eines ausgefallenen Knotens zu unterbrechen. Corosync kann die Clusterkommunikation ebenfalls über dieses Netzwerk als zusätzlichen Ring neben dem Frontend-Fabric abwickeln. Die Konfiguration von Corosync mit mehreren Ringen erhöht

die Redundanz für Cluster-Messaging und verringert das Risiko, dass ein einzelner Netzwerkausfall das Quorum unterbricht.

Lustre Clients

Ein Lustre-Client lädt das Client-Kernelmodul, stellt eine LNet-Verbindung zu OSS/MDS-Serverknoten her und bindet das Dateisystem ein, um Anwendungen einen einheitlichen, POSIX-konformen Namensraum bereitzustellen. Die Ein-/Ausgabe wird parallel auf die aktiven OSTs verteilt. Clientknoten befinden sich außerhalb eines Bausteins und nutzen das Dateisystem über das Frontend-LNet-Fabric. Client-Betriebssysteme sind für diese Lösung nicht in IMT aufgeführt; die "Lustre Support-Matrix" gibt getestete Client-Distributionen und Kernelversionen an, die von der bereitgestellten Lustre-Version unterstützt werden (zum Beispiel Red Hat Enterprise Linux 9, SUSE Linux Enterprise Server 15 und Ubuntu 24.04).

Clientknoten benötigen eine Verbindung zum selben LNet-Fabric und Netzwerktyp wie die OSS/MDS-Serverknoten. Falls erforderlich, kann ein LNet-Router separate LNet-Subnetze oder Fabrics überbrücken.

NetApp stellt vorkompilierte Lustre-Server-RPMs für Rocky Linux 9.8 und Red Hat Enterprise Linux 9.8 bereit. NetApp stellt keine vorkompilierten Client-Pakete bereit. Client-Pakete für das Client-Betriebssystem und den Kernel werden aus dem Lustre-Quellcode im "netapp-lustre Repository" erstellt.

Nach der Installation der Client-Pakete wird die optionale `lustre_client` Rolle in der "ansible-lustre Collection" verwendet, um die Client-Netzwerkschnittstellen und LNet zu konfigurieren, Multi-Rail-LNet bei Auswahl zu aktivieren, die Konnektivität zu validieren und eine persistente systemd-Mount-Unit zu verwalten. Die Rolle erstellt Lustre nicht. Die Installation der Client-Pakete erfolgt nur, wenn `lustre_client_packages` befüllt ist. Falls gewünscht, ist auch eine manuelle Konfiguration der Clients möglich. Schritt-für-Schritt-Verfahren finden sich unter "Die Lösung bereitstellen" und in der "Dokumentation der Rolle lustre_client".

Lustre mit NetApp E-Series Storage - Hardwarekomponenten

Diese Hardwareanforderungen für Server, NetApp E-Series Arrays, Storage-Layout und Netzwerk sind bei der Dimensionierung und Bestellung von Lustre mit NetApp E-Series Storage zu berücksichtigen. Für Softwarekomponenten siehe "Softwarekomponenten".

Serveranforderungen

Die Lösung basiert auf einem Bring-Your-Own-Server (BYOS) Modell. Jeder Baustein benötigt zwei OSS/MDS Serverknoten, die die folgenden Spezifikationen erfüllen oder übertreffen.

Knotenspezifikationen

Die folgende Tabelle listet die minimalen Serveranforderungen pro OSS/MDS-Knoten auf.

Komponente	Anforderung
Menge	2 Knoten pro Baustein
CPU	AMD EPYC oder Intel Xeon, 32 Kerne oder höher
Erinnerung	256 GB DDR5 (Minimum)
Netzwerk HCAs	6 Dual-Port 200Gb HCAs (siehe HCA Konnektivität)

Komponente	Anforderung
PCIe-Steckplätze	2× PCIe Gen5 x16 und 4× PCIe Gen5 x8 (siehe PCIe-Steckplatzanforderungen)
Bootlaufwerke	2× Laufwerke in RAID 1 (Software- oder Hardware-RAID empfohlen)

NetApp validierte die Lösung mit Lenovo ThinkSystem SR665 V3 Servern. Entsprechende Server anderer Hersteller werden unterstützt, sofern sie diese Anforderungen erfüllen.

HCA Konnektivität

Die folgende Tabelle listet die Anforderungen an HCA, LNet-Frontend-Port und NVMe-oF-Pfad pro OSS/MDS-Serverknoten auf.

HCAs pro Knoten	LNet Ports pro Lustre Knoten	NVMe-oF-Pfade pro Lustre Node	Beispiel HCA
6 (12 Anschlüsse)	4	8 insgesamt (4 pro Array)	MCX755106AS-HEAT (Dual port 200Gb PCIe gen5 Karte)

NetApp empfiehlt sechs HCAs pro Knoten, um die Bandbreite der EF80 Storage-Arrays zu maximieren.

PCIe-Steckplatzanforderungen

Jeder OSS/MDS-Serverknoten in einem EF80 Baustein verfügt über sechs Dual-Port-HCAs: zwei für LNet und vier für NVMe-oF. Die Steckplatzbreite bestimmt die für jede HCA verfügbare Bandbreite, daher ist die elektrische Breite jedes Steckplatzes und Risers zu prüfen und nicht nur die Breite der Karte allein. Eine Gen5 x16 Karte, die in einem Gen5 x8 Steckplatz installiert ist, läuft mit x8.

- **LNet HCAs:** In PCIe Gen5 x16 Steckplätzen installieren, um den vollen Frontend-Durchsatz zu erreichen.
- **NVMe-oF HCAs:** In PCIe Gen5 x8 oder PCIe Gen5 x16 Steckplätzen installieren.
- **NUMA-Balance:** Die HCAs gleichmäßig auf die beiden NUMA-Zonen aufteilen, sodass jede Zone zwei LNet-Schnittstellen und vier NVMe-oF-Schnittstellen umfasst.

Die folgende Tabelle listet die Mindestanzahl an Steckplätzen für jeden OSS/MDS-Serverknoten in einem EF80 Baustein.

Verkehrsart	HCAs pro Knoten	Mindestschlitzbreite	Slots pro NUMA-Zone
LNet	2	PCIe Gen5 x16	1
NVMe-oF	4	PCIe Gen5 x8	2

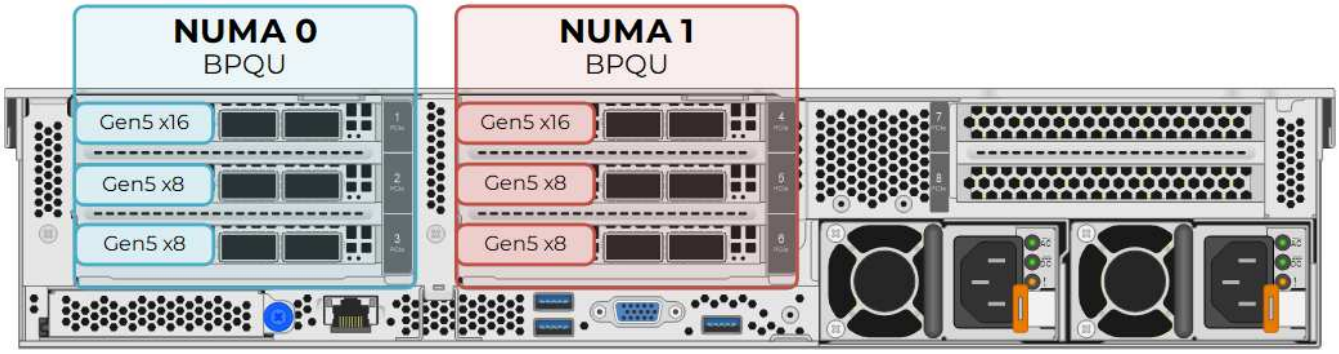
Optional ist eine Aufteilung der Ports eines Dual-Port-HCA möglich, sodass ein Port LNet-Datenverkehr und der andere NVMe-oF-Datenverkehr verarbeitet. Alle vier dieser HCAs werden in PCIe Gen5 x16-Steckplätzen installiert, sodass jeder LNet-Port die volle Durchsatzleistung erreicht. Anschließend werden zwei weitere HCAs in Gen5 x8- oder Gen5 x16-Steckplätzen für die verbleibenden NVMe-oF-Ports hinzugefügt.

▼ **Beispielhafte Riser-Konfigurationen für Lenovo ThinkSystem SR665 V3**

Beide der folgenden Steigleitungskombinationen erfüllen die Anforderungen an die Steckplätze für einen EF80 Baustein.

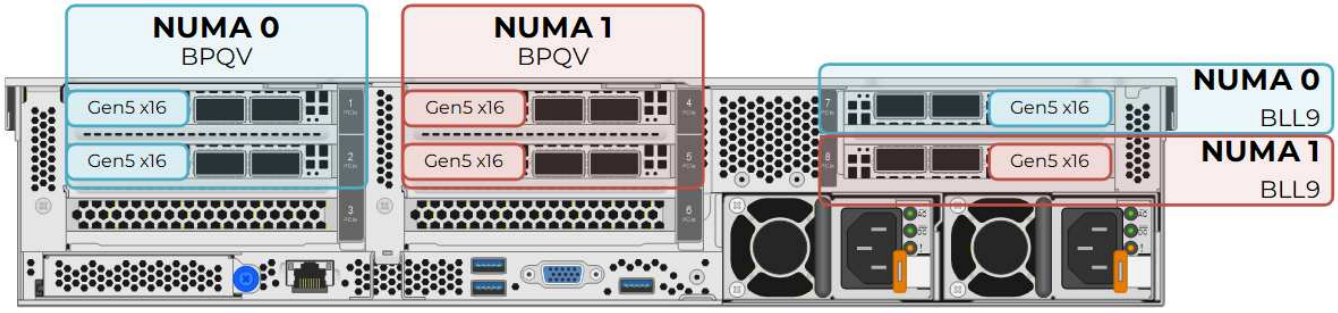
Mindeststeckplatzbreiten

Eine BPQU-Riserkarte in den Riser-Positionen 1 und 2 installieren. Jede BPQU-Riserkarte bietet einen PCIe Gen5 x16 Slot und zwei PCIe Gen5 x8 Slots. Zusammen bieten die Riserkarten zwei Gen5 x16 Slots für LNet und vier Gen5 x8 Slots für NVMe-oF, gleichmäßig auf die NUMA-Zonen verteilt.



Alle Gen5 x16 Steckplätze

Eine BPQV Riserkarte ist in den Riser-Positionen 1 und 2 zu installieren und eine BLL9 Riserkarte in Riser-Position 3. Jede BPQV Riserkarte stellt zwei PCIe Gen5 x16-Steckplätze bereit. Die BLL9 Riserkarte stellt Steckplatz 7 in NUMA-Zone 0 und Steckplatz 8 in NUMA-Zone 1 bereit, beide PCIe Gen5 x16. Diese Kombination stellt sechs Gen5 x16-Steckplätze bereit, drei pro NUMA-Zone, und unterstützt die Aufteilung des LNet- und NVMe-oF-Datenverkehrs auf die Ports derselben HCA.



EF80 Baustein

Die EF80 ist die validierte Standardplattform für diese Lösungsversion.

Array-Spezifikationen

Die folgende Tabelle enthält die EF80 Array-Spezifikationen für einen Baustein (zwei Arrays).

Komponente	Spezifikation
Modell	NetApp EF80
Formfaktor	2U Basisgehäuse, 24 interne NVMe SSD-Steckplätze
Controller	Dual Controller (A und B)
Laufwerke	24× NVMe SSDs pro Array
E/A-Konnektivität	8 × 200Gb NVMe/IB oder NVMe/RoCE Host-Ports pro Array im validierten Lustre Design

Die EF80 Plattform unterstützt bis zu zwölf Host-Ports pro Array, wenn in jedem Controller drei Zwei-Port-Host-I/O-Module installiert sind. Das validierte Lustre Design verwendet Host-I/O-Module in den Steckplätzen 1 und 2 für acht Host-Ports pro Array.

Jedes EF80 Array nutzt außerdem das dedizierte I/O-Modul in Steckplatz 4 für die Inter-Controller-Spiegelung. Controller A Port 4a wird mit Controller B Port 4a verbunden, und Controller A Port 4b mit Controller B Port 4b. Diese Verbindungen dienen der Cache-Spiegelung und dem I/O-Transport und werden nicht für den Host-NVMe-oF-Datenverkehr verwendet. Siehe "Die Inter-Controller-Spiegelungsverbindungen EF50 und EF80 verkabeln".

Die vollständigen technischen Daten der EF-Series für alle Modelle sind im "NetApp EF-Series All-Flash-Array Datenblatt" zu finden.

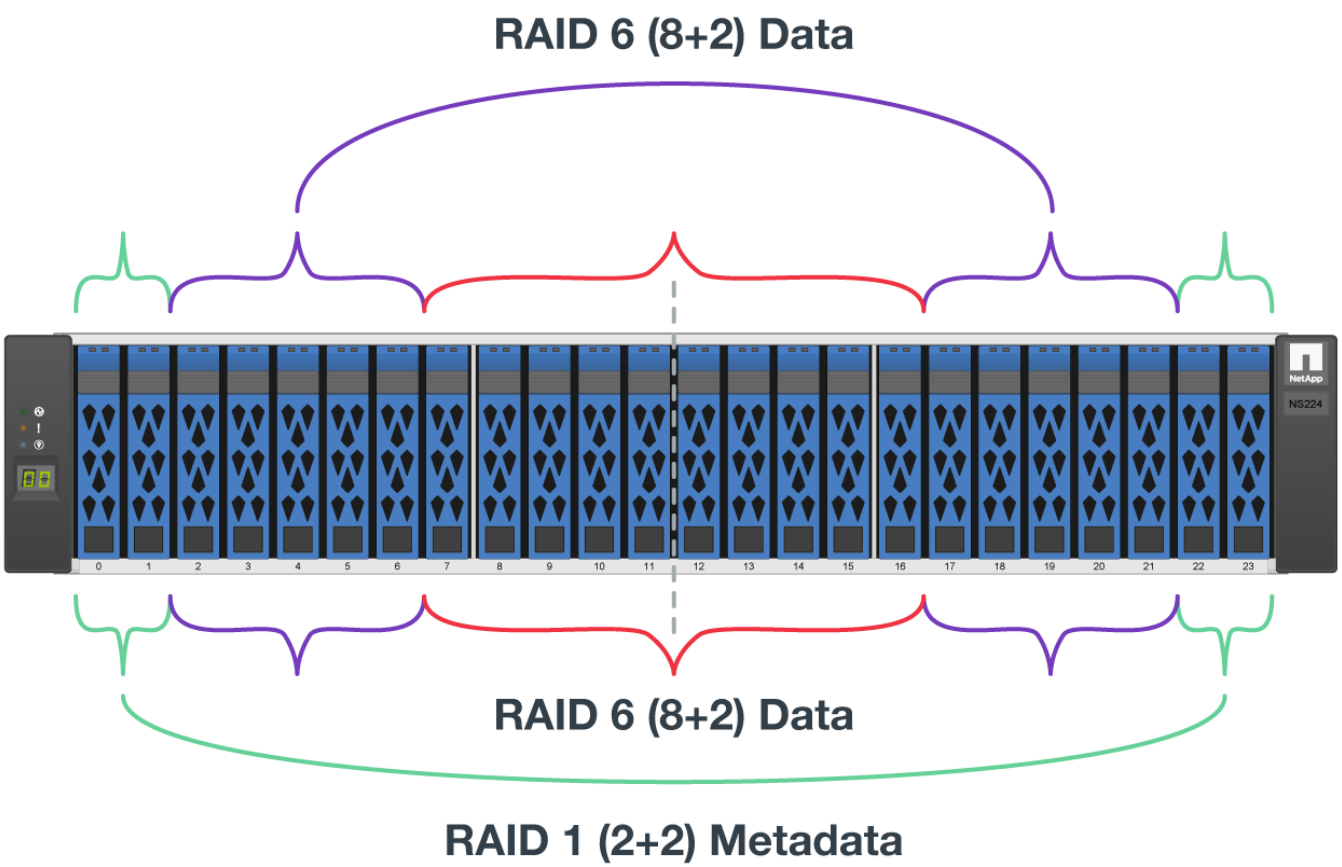
Laufwerksanordnung (24 Laufwerke pro Array)

Jedes EF80 Array in einem Baustein nutzt alle vierundzwanzig NVMe-Laufwerkssteckplätze:

- 4 Laufwerke im RAID 1 Verbund für MGS/MDT Storage (gemeinsame Volume-Gruppe oder DDP-Zuordnung)
- 10× Laufwerke in RAID 6 für OST-Speicher (erster OST-Pool)
- 10 Laufwerke in RAID 6 für OST-Speicher (zweiter OST-Pool)

Dieses Layout gilt, wenn 3,84-TB-, 7,68-TB- oder 15,3-TB-Laufwerke mit herkömmlichen Volume-Gruppen verwendet werden. Bei Verwendung von 30,7-TB- oder 61,4-TB Capacity Flash (QLC) Laufwerken wird ein einzelner Dynamic Disk Pool über alle 24 Laufwerke bereitgestellt und innerhalb dieses Pools werden RAID 1 Volumes für MGS- und MDT-Volumes erstellt, während für OSTs standardmäßig RAID 6 Volumes verwendet werden.

Für diese Lösung werden derzeit keine Festplatten mit 1,92 TB Speicherkapazität empfohlen. Es sollte eine der oben aufgeführten validierten Festplattenkapazitäten verwendet werden.



Volume- und Zielanzahl (Basis-Baustein)

Die folgende Tabelle listet die MGS-, MDT- und OST-Zähler für einen Basis-Baustein auf.

Zieltyp	Anzahl pro BB	RAID / Pool	Hinweise
MGS	1	RAID 1	Nur Basisbaustein; Array 1
MDT	8	RAID 1	4 pro Array
OST	32	RAID 6 (TLC) oder DDP (QLC)	16 pro Array

Die folgenden Richtlinien zur Volumendimensionierung dienen als Ausgangspunkt in der Ansible-Inventur.

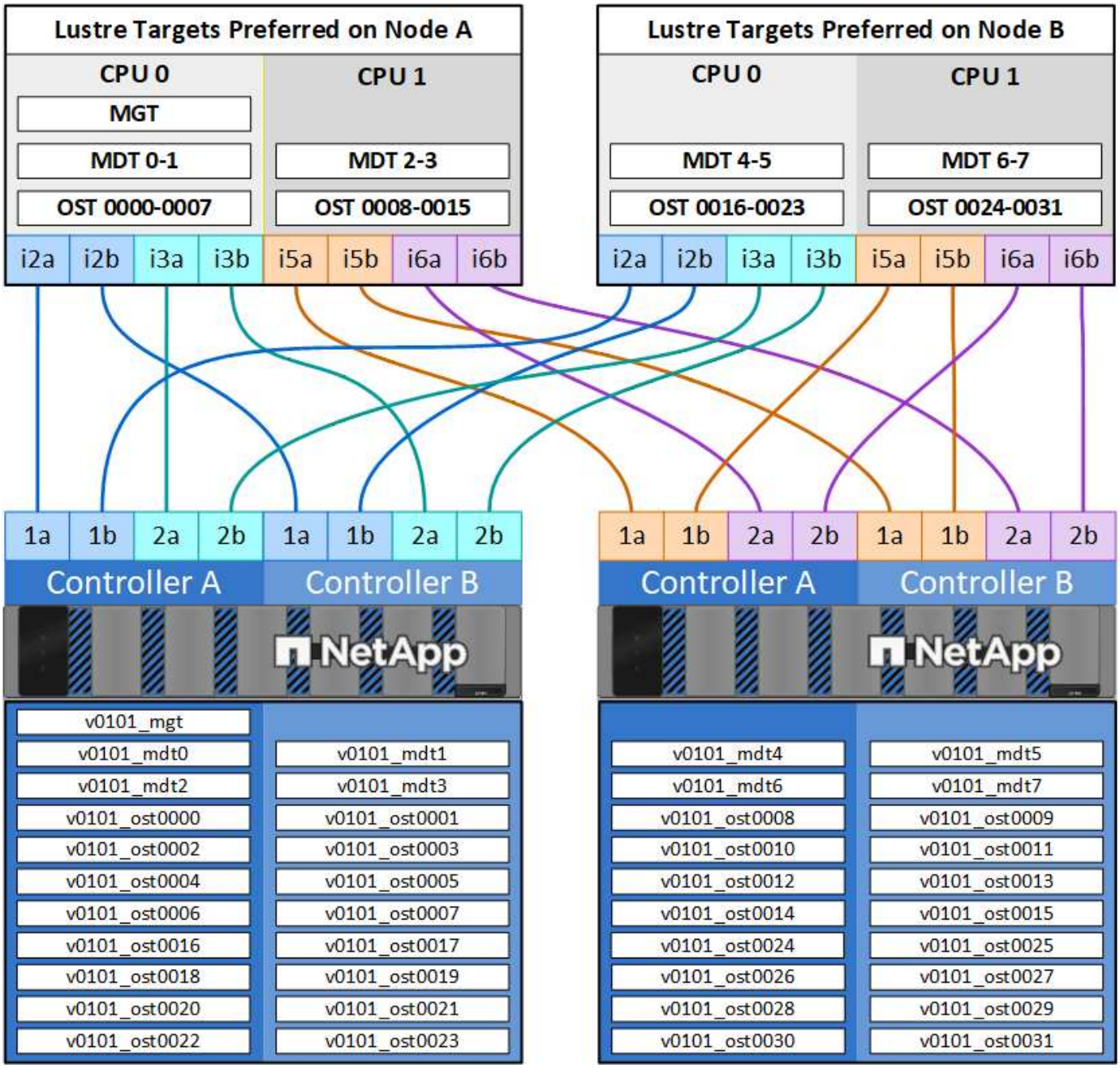
Volumen	Empfohlene Größe	Hinweise
MGS	5–10 GiB	Nur Konfigurationsdaten
MDT	RAID 1 Kapazität ÷ MDT-Anzahl	Jeweils etwa 1–2 TiB (typisch)
OST	RAID 6 oder DDP Kapazität ÷ OST-Anzahl pro Pool	Skaliert mit der Laufwerkskapazität; siehe "Größenberatung"

Primäre Baustein-Volume-Verteilung

Die folgende Abbildung zeigt die bevorzugte Platzierung der Lustre-Ziele und die NVMe-oF-Konnektivität

zwischen den OSS/MDS-Serverknoten und E-Series Arrays in einem EF80 Baustein. Das Diagramm kennzeichnet das Management-Ziel als **MGT** (management target); ein MGT hostet den MGS (management server) Dienst, der an anderer Stelle in diesem Dokument erwähnt wird.

Verteilung des Basisbausteins (EF80)



Die folgende Tabelle zeigt, wie die Volumes auf die beiden Arrays verteilt sind und welchen Server jedes Ziel bevorzugt.

Zieltyp	Array 1	Array 2	Server 1	Server 2	Hinweise
MGS	1	0	1	0	Nur Basisbaustein
MDT	4	4	4	4	Jeder Server bevorzugt 2 MDTs aus jedem Array

Zieltyp	Array 1	Array 2	Server 1	Server 2	Hinweise
OST	16	16	16	16	8 pro RAID 6 Pool × 2 Pools pro Array oder äquivalente DDP-Zuweisung
Gesamtvolumen	**21**	**20**	**21**	**20**	

Auswahl des Speicherpools: TLC vs QLC

Wählen Sie den E-Series Pooltyp anhand der NVMe-Laufwerkskapazität in den Arrays:

- **3,84-TB-, 7,68-TB- und 15,3-TB-Laufwerke:** Für OST-Volumes werden **RAID 6 Volume-Gruppen** und für MGS- sowie MDT-Volumes **RAID 1 Volume-Gruppen** verwendet, alternativ kann DDP für den gemeinsam genutzten Laufwerkspool eingesetzt werden.
- **30,7-TB- und 61,4-TB-Kapazität Flash (QLC)-Laufwerke:** Es ist ausschließlich die Verwendung von **Dynamic Disk Pools (DDP)** vorgesehen. RAID 1-Volumes sind innerhalb des DDP für MGS- und MDT-Speicher zu erstellen. RAID 6-Volumes für OST-Speicher werden aus dem DDP erstellt.

Für Schätzungen der nutzbaren Kapazität pro Baustein nach Laufwerksgröße und -anordnung siehe "Größenberatung".

Netzwerkanforderungen

Backend (NVMe-oF)

- NVMe/InfiniBand oder NVMe/RoCE zwischen jedem OSS/MDS-Knoten und jedem E-Series-Array
- MTU 9000 auf NVMe/RoCE Backend-Schnittstellen (typisch)
- Acht Pfade von jedem Lustre-Knoten zu den Speicherarrays (vier zu jedem Array)
- EF80 verwendet sechs HCAs pro Knoten (i2, i3, i5 und i6 für NVMe-oF; i1 und i4 für LNet). Knoten A wird mit Controller-Ports verkabelt, deren Bezeichnungen auf `a` enden, wie zum Beispiel `1a` und `2a`. Knoten B wird mit Controller-Ports verkabelt, deren Bezeichnungen auf `b` enden, wie zum Beispiel `1b` und `2b`. Siehe "EF80 Six-HCA-Backend-Verkabelung".

Frontend (LNet)

- IPoIB oder RoCE für LNet (`@o2ib` Netzwerktyp)
- MTU 9000 auf RoCE Frontend-Schnittstellen (typisch)
- Multi-rail LNet wird empfohlen, wenn vier Frontend-Ports verfügbar sind (EF80: i1 und i4 HCAs, beide Ports)
- Dedizierte InfiniBand oder RoCE Switch-Fabric zur Verbindung der OSS/MDS-Serverknoten und Lustre-Clients
- Verlustfreies RoCE (PFC) wird für RoCE-Fabrics empfohlen.

Management

- Out-of-band-Managementnetzwerk für Server-BMCs und Array-Management-Ports
- Dediziertes Corosync Netzwerk oder gemeinsam genutzte Frontend Infrastruktur (standortabhängig)

Lustre mit NetApp E-Series Storage - Softwarekomponenten

Die Software-Stack und die Ansible-Automatisierung, die Lustre mit NetApp E-Series Storage bereitstellen, werden dargestellt. Für Server, Storage und Netzwerke siehe "Hardwarekomponenten".

Baustein-Software

Die folgende Tabelle listet die von E-Series Arrays und OSS/MDS-Serverknoten verwendeten Softwarekomponenten auf. Die "NetApp Interoperabilitätsmatrix Tool (IMT)" ist die maßgebliche Quelle für unterstützte Versionen und Komponentenkombinationen. Es sollte nach **E-Series Lustre** gesucht und vor der Bereitstellung bestätigt werden, dass die Versionen von Betriebssystem, Lustre, SANtricity OS, DOCA-OFED, HCA Firmware und HA-Paket korrekt sind.

Komponente	Funktion
SANtricity OS	Läuft auf jedem E-Series Array und bietet SANtricity System Manager, Dual-Active-Controller Hochverfügbarkeit mit automatisiertem I/O-Pfad-Failover, automatischem Lastausgleich und Pfadüberwachung, Dynamic Disk Pools und traditionelles RAID, automatischen Laufwerkswiederaufbau, gespiegelten Cache-Schutz, Data Assurance, proaktive Laufwerkszustandsüberwachung sowie Online-Software-, Firmware- und Konfigurationsänderungen.
Lustre	Clients wird ein einziger POSIX-konformer Namensraum bereitgestellt, während Metadaten und Dateidaten für den parallelen Zugriff auf mehrere Ziele verteilt werden. Der Management Server (MGS) speichert die Dateisystemkonfiguration, Metadatenziele (MDTs) verwalten den Namensraum und Objektspeicherziele (OSTs) speichern Dateidaten auf E-Series Volumes. Diese Trennung ermöglicht die Skalierung von Leistung und Kapazität durch Hinzufügen von Bausteinen.
Red Hat Enterprise Linux (RHEL) oder Rocky Linux	Bietet den Kernel, Systemdienste und das Paketframework für Lustre-Zieldienste, HA-Clustering sowie Speicher- und Netzwerkkonnektivität. Die unterstützten Betriebssystem-Repositories enthalten außerdem Pacemaker, Corosync, Fence Agents, NVMe-oF und Multipath-Pakete, die gemeinsam als Lösungsstack getestet werden.
Hochverfügbarkeits-Cluster-Services (Pacemaker, Corosync und fence agents)	Diese Dienste gewährleisten gemeinsam eine koordinierte Hochverfügbarkeit für Lustre Ziele auf OSS/MDS Serverknoten. Sie verwalten die Clusterzugehörigkeit und das Quorum, überwachen Ressourcen, orchestrieren ein geordnetes Failover und isolieren nicht reagierende Knoten, bevor gemeinsam genutzte Ziele an anderer Stelle aktiviert werden. Diese Koordination verhindert Split-Brain-Zugriffe auf E-Series Volumes und schützt die Dateisystemintegrität bei Ausfällen.
NVIDIA DOCA-OFED	Bietet HCA-Treiber, RDMA-Bibliotheken und Verwaltungsprogramme für InfiniBand und RoCE-Fabrics. Derselbe Software-Stack unterstützt mit niedriger Latenz Backend-NVMe-oF-Zugriff auf E-Series Arrays und hochperformante Frontend-LNet-Kommunikation mit Lustre Clients.

NetApp bietet vorkompilierte Lustre Server RPMs für Rocky Linux 9.8 und Red Hat Enterprise Linux 9.8.

Ansible Automatisierung

Lustre mit NetApp E-Series Storage wird mithilfe von Ansible-Automatisierung von einem Kontrollknoten mit Verwaltungszugriff auf die E-Series Arrays und OSS/MDS-Serverknoten bereitgestellt und implementiert. Ein Ansible-Inventar bildet die gewünschte Lösung ab, einschließlich Arrays, Server, Speicherzuweisungen, Netzwerkschnittstellen und HA-Cluster-Ressourcen.

Die NetApp E-Series Lustre Ansible Collection orchestriert die End-to-End-Bereitstellung mithilfe der SANtricity und Host Collections. Gemeinsam stellen diese Collections E-Series Storage bereit und ordnen ihn zu, bereiten das Serverbetriebssystem vor, konfigurieren NVMe-oF- und LNet-Konnektivität, stellen Lustre Targets bereit und erstellen Pacemaker-Ressourcen. Diese Automatisierung macht die Konfiguration wiederholbar und vereinfacht das Hinzufügen von Bausteinen zu einem bestehenden Dateisystem.

Die folgende Tabelle fasst die wichtigsten Softwarepakete zusammen, die auf dem Ansible-Kontrollknoten verwendet werden. Die "NetApp E-Series Lustre Ansible Kollektion" für die aktuellen Paketabhängigkeiten.

Paket	Abhängigkeit oder Zweck
Python	Stellt die Laufzeitumgebung für Ansible und die erforderlichen Python-Module bereit.
ansible-core	Erfordert Python und führt die NetApp E-Series Ansible-Sammlungen aus.
Zusätzliche Python Pakete	`cryptography`, `ipaddr`, `netaddr`, `passlib`, `resolvelib`, `jinja2` und `pyyaml`
NetApp E-Series Lustre Ansible Kollektion	Konfiguriert Lustre, LNet, NVMe-oF Hostkonnektivität und Pacemaker-Ressourcen.
NetApp E-Series SANtricity Ansible-Sammlung	Konfiguriert E-Series Arrays, Speicherpools, Volumes und Host-Mappings über die SANtricity Web Services API.
NetApp E-Series Host Ansible Collection	Bereitet OSS/MDS-Serverknoten für die LNet- und NVMe-oF-Konnektivität zu E-Series Arrays vor.

Lustre Client Software

Die Lustre Clientsoftware ermöglicht einem System den Zugriff auf das parallele Dateisystem über das Frontend LNet Fabric. Jeder Client benötigt Lustre Kernelmodule und Lustre Client Dienstprogramme, die sowohl mit dem verwendeten Kernel als auch mit der auf den Servern installierten Lustre Version kompatibel sind.

Verwenden Sie das "Lustre Support-Matrix", um ein kompatibles Client-Betriebssystem und einen Kernel für Ihre Lustre-Version zu finden. NetApp stellt keine vorkompilierten Lustre-Clientpakete bereit. Clientpakete für das Client-Betriebssystem und den Kernel werden aus dem von NetApp bereitgestellten Lustre-Quellcode im "netapp-lustre Repository" erstellt. Nachdem die Lustre-Clientpakete installiert wurden, kann die `lustre_client` Rolle Netzwerkschnittstellen, LNet und eine persistente systemd-Mount-Unit für das Lustre-Dateisystem konfigurieren.

Informationen zu den Schritten zur Clientbereitstellung finden sich unter "Die Lösung bereitstellen".

Lustre mit NetApp E-Series Storage – Größenempfehlungen

Lustre mit NetApp E-Series Storage Bausteinen nach Kapazitäts- und Metadatenbedarf

dimensionieren, den Zeitpunkt für die Erweiterung der Kapazität bestimmen und die Faktoren betrachten, die die Leistung beeinflussen.

Kapazitätsauslegung

Ein Baustein mit zwei EF80-Arrays bietet das folgende Lustre-Target-Layout. Bevorzugte Target-Platzierung und NVMe-oF-Pfade sind in "Primäre Baustein-Volume-Verteilung" in "Hardwarekomponenten" dargestellt.

Komponente	Anzahl	Typische Volumengröße (pro Ziel)
MGS	1	5-10 GiB
MDT	8	Die Größe variiert je nach Laufwerkskapazität und RAID-Layout
OST	32	Die Größe variiert je nach Laufwerkskapazität und RAID-Layout

Jedes EF80 Array verwendet vierundzwanzig NVMe-Laufwerke. Unterstützte Kapazitäten sind 3,84 TB, 7,68 TB und 15,3 TB mit herkömmlichen Volume-Gruppen (RAID 1 für MGS/MDT und RAID 6 für OST) oder DDP sowie 30,7 TB oder 61,4 TB Capacity Flash (QLC) Laufwerke ausschließlich mit DDP. 1,92 TB Laufwerke werden für diese Lösung derzeit nicht empfohlen. Für Informationen zur Laufwerkssteckplatzbelegung und Poolauswahl siehe "Hardwarekomponenten".

Die folgende Tabelle zeigt die ungefähre nutzbare Kapazität eines EF80 Bausteins (zwei Arrays, 48 Laufwerke). Die nutzbare Kapazität des Arrays entspricht nicht der endgültigen nutzbaren Kapazität des Lustre-Dateisystems. MDT-Inode-Zuweisung, Journale, Überprovisionierung und der Anteil des Inventarvolumens reduzieren die für Anwendungen verfügbare Kapazität.

Laufwerkskapazität	Layout (pro Array)	Ungefähr nutzbare Kapazität (Baustein)
3,84 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	138,08 TB
3,84 TB	DDP (24 Laufwerke, 2 reserviert)	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	Nur DDP	1.070,35 TB
61,4 TB	Nur DDP	2.162,70 TB

Metadaten und Daten separat dimensionieren, anschließend die Volume-Größen im Ansible-Inventar festlegen. Die Bereitstellungsvorlagen formatieren Ziele mit ldiskfs; MDT verwendet das standardmäßige Inode-Verhältnis von Lustre, und OST-Vorlagen setzen `-i 4096`.

- **Metadaten (MDT):** MDTs anhand der erwarteten Dateianzahl dimensionieren. Für das Wachstum etwa die doppelte erwartete Dateianzahl einplanen. Ein zu klein dimensioniertes MDT kann die Erstellung neuer

Dateien verhindern, selbst wenn noch OST-Kapazität vorhanden ist.

- **Daten (OST):** OSTs werden anhand der nutzbaren Kapazität und des Durchsatzbedarfs dimensioniert.
- **MGS:** Ein kleines Management-Target (5-10 GiB) sollte ausschließlich für die Dateisystemkonfiguration verwendet werden.

Passen Sie die MDT- und OST-Volumegrößen im Inventar an die ausgewählte Laufwerkskapazität an. Eine Änderung von `format_options.mkfsoptions` in `eseries_lustre_filesystem_mdt` oder `eseries_lustre_filesystem_ost` erfolgt nur, wenn ein Standort ein anderes Inode-Verhältnis benötigt. Hintergrundinformationen zu Inode-Verhältnissen stehen im "Lustre Wiki: Lustre Tuning" zur Verfügung. Hinweise dazu, wann Bausteine hinzugefügt werden sollten, finden sich unter Skalierung.

Skalierung

Bausteine können hinzugefügt werden, wenn die Kapazität oder die Metadatenlast dies erfordert:

- **Nur OST:** Dateianzahl und Metadatenlast liegen innerhalb der vorhandenen MDT-Kapazität, und es wird mehr Datenkapazität oder ein höherer Gesamtdurchsatz benötigt. 32 OSTs und zwei OSS/MDS-Serverknoten werden hinzugefügt.
- **MDT+OST:** Die Anzahl der Dateien oder die Metadatenrate nähert sich der MDT-Kapazität, oder es wird sowohl ein höherer Metadatendurchsatz als auch eine höhere Datenkapazität benötigt. Es werden 8 MDTs und 32 OSTs hinzugefügt.

Mit `mdt.*.md_stats` auf bestehenden MDTs lässt sich bestätigen, ob die Metadaten ausgelastet sind. Jeder Pacemaker/Corosync Cluster ist auf fünf Bausteine (zehn OSS/MDS Serverknoten) zu begrenzen. Bei größeren Bereitstellungen werden die Bausteine gleichmäßig auf mehrere HA-Cluster verteilt. Siehe "Lösungsarchitektur" für Skalierungsregeln.

Das folgende Beispiel zeigt ein Dateisystem mit einem Basis-Baustein plus einem reinen OST-Baustein in einem HA Cluster.

Baustein	Typ	Server	Arrays	MGS	MDT	OST
BB1	Basis	2	2	1	8	32
BB2	nur OST	2	2	0	0	32
Gesamt		4	4	1	8	64

BB2 registriert seine OST-Ziele beim MGS in BB1 mithilfe von `mgsnode=`. Die OST-Indizes für BB2 werden global zugewiesen (zum Beispiel OST0032 bis OST0063), sodass kein Index mit BB1 in Konflikt steht.

Performance

Die formale Validierung nutzt IOR, mdtest und fio von Lustre Clients, um den relativen Durchsatz, die IOPS und das Metadatenverhalten zu charakterisieren und HA Failover zu validieren. Diese Tests veröffentlichen keine garantierten Leistungswerte. Synthetische Benchmarks messen das optimale Verhalten und spiegeln möglicherweise nicht die tatsächliche Anwendungsleistung wider.

Die Leistung skaliert annähernd mit der Anzahl der Bausteine. Die erzielten Ergebnisse hängen vom Laufwerkstyp und Pool-Layout, der Anzahl der NVMe-oF-Pfade, der LNet-Fabric und der Anzahl der Clients sowie vom Verhältnis von Daten- zu Metadaten-I/O ab.

Wenden Sie sich an Ihr NetApp Account-Team, um Empfehlungen zur Dimensionierung und Leistung speziell

für Ihre Arbeitslast zu erhalten.

Informationen zum Laufwerkslayout und zur Anzahl der Volumes finden sich unter "Hardwarekomponenten". Die Bereitstellungsschritte sind unter "Die Lösung bereitstellen" aufgeführt. Hinweise zur Bestandsaufnahme auf Feldebene finden sich unter "Das Lustre Ansible-Inventar anpassen".

Die Lösung bereitstellen

Lustre mit NetApp E-Series Storage bereitstellen

Lustre mit NetApp E-Series Storage wird bereitgestellt, indem die Aufgaben für Hardware, Umgebungsvorbereitung, Inventarisierung, Ansible-Bereitstellung und Verifizierung in der vorgegebenen Reihenfolge abgeschlossen werden.

Voraussetzungen

Vor Beginn der Bereitstellung ist Folgendes sicherzustellen:

- Sie haben die Vorgaben zur Dimensionierung und zu den Bausteinen in "Größenberatung" befolgt und Hardware ausgewählt, die "Hardwarekomponenten" entspricht.
- Sie haben bestätigt, dass die Kombination aus Server, HCA, E-Series, Betriebssystem, Lustre, SANtricity OS und DOCA-OFED für **E-Series Lustre** in der "NetApp Interoperabilitätsmatrix-Tool" aufgeführt ist.
- Alle OSS/MDS Serverknoten und E-Series Arrays für die geplanten Bausteine sind vor Ort und bereit zum Einbau.
- Sie verfügen über Anmeldeinformationen und Netzwerkzugriff für Server-BMCs, Array-Management-Ports und den Host, der als Ansible Kontrollknoten fungiert.

> NetApp unterstützt die Bereitstellung des Lustre HA-Clusters und der Lustre-Clients mit dem "NetApp E-Series Lustre Ansible Kollektion". Lustre-Ziele, LNet, NVMe-oF-Hostkonnektivität oder Pacemaker-Ressourcen sollten nicht manuell konfiguriert werden.

Bereitstellungs-Workflow

Die folgenden Aufgaben sind in der angegebenen Reihenfolge auszuführen:

1. "Rack- und Kabel-Lustre-Hardware".

   Die Lustre-Server und EF80 Arrays werden in Racks montiert, anschließend erfolgt die Verkabelung der Backend-NVMe-oF- und Frontend-LNet-Fabrics.

2. "Die Lustre-Bereitstellungsumgebung vorbereiten".

   Die Arrays und Server werden konfiguriert, anschließend werden Ansible und dessen Abhängigkeiten auf dem Kontrollknoten installiert.

3. "Das Lustre Ansible-Inventar anpassen".

   Die Vorlage für das Fabric der Website auswählen und die Server, Arrays, Netzwerke, Bausteine und Anmeldeinformationen eintragen.

4. "Bereitstellung des Lustre HA Clusters und der Clients".

Das Serverbetriebssystem wird vorbereitet, NVIDIA DOCA-OFED installiert, das Hauptbereitstellungs-Playbook ausgeführt und die Lustre-Clients konfiguriert.

5. "Lustre Bereitstellung überprüfen und erweitern".

   Vor dem Produktiveinsatz sind der Zustand des Clusters, die Mounts und die Kapazität zu bestätigen. Das bestehende Inventar wird erweitert, wenn das Dateisystem einen weiteren Baustein benötigt.

Verwandte Informationen

- "NetApp E-Series Lustre Ansible Kollektion"
- "NetApp Lustre-Quellcode-Repository"
- "HA Administrationshandbuch"
- "Leitfaden zur Leistungsoptimierung"
- "Lustre Client-Bereitstellungsvorlagen"

Rack und Kabel Lustre Hardware

Die Lustre-Server und NetApp EF80 Arrays werden in Racks montiert, anschließend werden die Backend-NVMe-oF- und Frontend-LNet-Fabrics für jeden Baustein verkabelt.

Bevor Sie beginnen

Die HCA-Anzahl, die PCIe-Steckplatzanforderungen, die Pfadanforderungen und die MTU-Richtlinien in "Hardwarekomponenten" sollten geprüft werden.

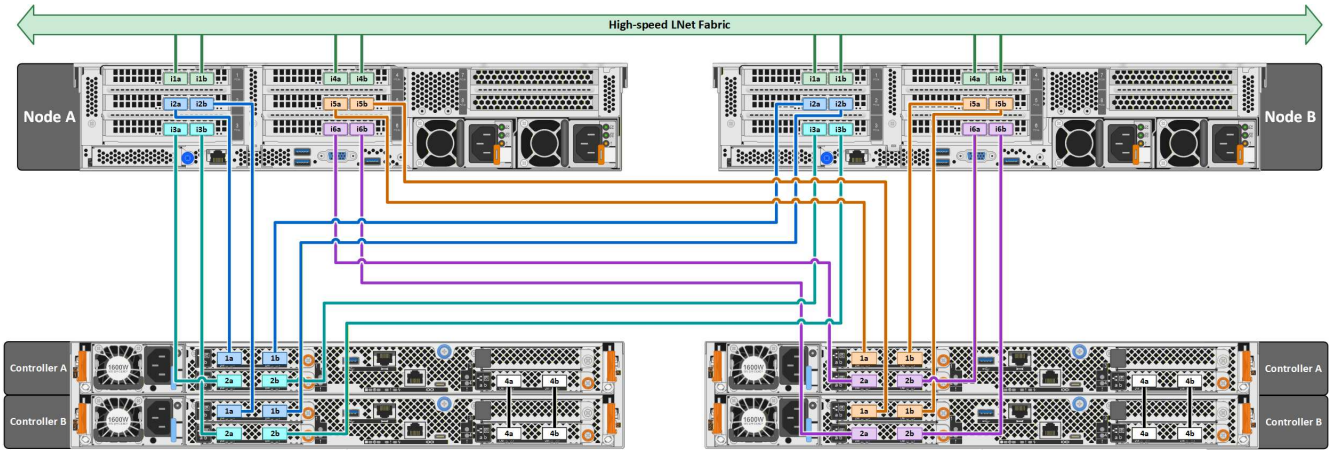
Schritte

1. Alle Lustre-Server und E-Series-Speicherarrays werden gemäß der Baustein-Topologie installiert und verkabelt.

2. Für jedes EF80 Array wird Controller A Port 4a mit Controller B Port 4a und Controller A Port 4b mit Controller B Port 4b verbunden. Diese dedizierten Verbindungen ermöglichen die Inter-Controller-Spiegelung und werden nicht für den Host-Datenverkehr verwendet. Siehe "Die Inter-Controller-Spiegelungsverbindungen EF50 und EF80 verkabeln".

3. Das Backend NVMe-oF Netzwerk zwischen Servern und Arrays wird gemäß der EF80 sechs-HCA-Topologie verkabelt.

EF80 sechs-HCA Backend-Verkabelung (RoCE oder InfiniBand)



4. Die vier Frontend-LNet-Schnittstellen (die i1 und i4 HCAs) auf jedem Lustre Server sowie die Lustre Clients werden mit dem dedizierten InfiniBand oder RoCE Switch Fabric verbunden.
5. Bei Verwendung von RoCE wird das Fabric für verlustfreien Betrieb mit priorisierter Flusssteuerung (PFC) konfiguriert. Die Ansible Playbooks konfigurieren verlustfreies RoCE auf allen Schnittstellen im Baustein.

Nachdem Sie fertig sind

"Vorbereitung der Server, Arrays und des Ansible-Kontrollknotens".

Die Lustre-Bereitstellungsumgebung vorbereiten

Der Netzwerkzugriff und die Basiseinstellungen werden auf jedem EF80 Array und Lustre Server konfiguriert, anschließend wird ein Ansible Kontrollknoten für die Bereitstellung der Lösung vorbereitet.

E-Series Arrays vorbereiten

Schritte

1. Die Managementoberfläche auf jedem Controller wird konfiguriert und es wird überprüft, ob SANtricity System Manager erreichbar ist.
2. Legen Sie den Array-Namen und die administrativen Anmeldeinformationen fest, die das Ansible-Inventar verwendet.
3. Bestätigen Sie, dass sich das Array in einem optimalen Zustand befindet und dass alle Laufwerke vorhanden sind und korrekt melden.

Anweisungen zur Konfiguration der Managementverbindung sind im "E-Series Dokumentation" zu finden.

Lustre-Server vorbereiten

Schritte

1. Der Baseboard Management Controller (BMC) wird auf jedem Server für den Out-of-Band-Zugriff konfiguriert, und die UEFI-Systemeinstellungen werden für maximale Leistung festgelegt.

2. Ein unterstütztes Betriebssystem (RHEL oder Rocky Linux) wird installiert, anschließend werden der Hostname und der Managementnetzwerkport konfiguriert.
3. Alle Firmware- und Softwarepakete werden so vorbereitet, dass sie die Anforderungen für die **E-Series Lustre** Lösung erfüllen, wie in "Softwarekomponenten" beschrieben.

Den Ansible-Kontrollknoten vorbereiten

Ein virtuelles oder physisches Linux-System mit Managementnetzwerkzugriff auf alle Lustre-Server und E-Series-Arrays wird als Ansible-Kontrollknoten verwendet.

Die folgenden Schritte beziehen sich auf Ubuntu 24.04. Anweisungen für eine andere Linux-Distribution finden sich im "Ansible Installationsdokumentation".

Schritte

1. Python, pip und das Python Virtual Environment-Paket installieren:

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

2. Erstellen und Aktivieren einer virtuellen Python-Umgebung:

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

3. Ansible und die erforderlichen Python-Pakete werden in der aktivierten virtuellen Umgebung installiert:

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

4. Die NetApp E-Series Lustre Ansible Collection wird installiert. Mit der Installation der Collection werden auch die abhängigen Collections, einschließlich der SANtricity und Host Collections, installiert:

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

5. Die installierten Versionen von Ansible, Python und der Collection können überprüft werden:

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

6. Passwortloses SSH wird vom Kontrollknoten zu jedem Lustre Server konfiguriert. `<ip_or_hostname>` ist durch die Management-IP-Adresse oder den Hostnamen eines Servers zu ersetzen:

```
ssh-keygen
```

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

Wiederholen `ssh-copy-id` für jeden Lustre Server.

> Passwortloses SSH sollte für die E-Series-Arrays nicht konfiguriert werden. Die Verwaltung der Arrays durch Ansible erfolgt über die SANtricity Web Services API unter Verwendung der im Inventar definierten Anmeldeinformationen.

Nachdem Sie fertig sind

"Das Ansible Inventar anpassen".

Das Lustre Ansible Inventory anpassen

Eine Arbeitskopie einer EF80-Bereitstellungsvorlage erstellen und deren Inventar für Ihre Server, Arrays, Netzwerke und Lustre Bausteine anpassen.

Das ausgewählte Inventar bleibt im zugehörigen Fabric- und Plattformverzeichnis. Das freigegebene `playbooks` Verzeichnis und die `passwords.yml` Datei bleiben im Stammverzeichnis von `building_blocks` erhalten.

Eine Bereitstellungsvorlage auswählen

Beginnen Sie mit der Vorlage, die zur Netzwerkstruktur der Website passt:

- **InfiniBand:** "IB_H2_IB_DAC_A2_EF80/lustre_Baustein_cluster_1"
- **RoCE:** "ROCE_H2_ROCE_DAC_A2_EF80/lustre_baustein_cluster_1"

Standortspezifische Einstellungen konfigurieren

Die standortspezifischen Werte werden im Inventar zusammen mit den Einstellungen für das lokale Repository und die udev-Regeln festgelegt. Zeilenverweise in der Tabelle beziehen sich auf die RoCE EF80 Vorlagendateien im `ansible-lustre release/1.0.0` Tag. Die InfiniBand Vorlage verwendet dieselben Variablennamen und dieselbe Struktur, mit Ausnahme der LNet Schnittstellenvariablen, die in der Tabelle aufgeführt ist.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
Dateisystemname	`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"	Clusterweit	Verschachtelt unter `format_options` für die MGT-, MDT- und OST-Zuordnungen. Maximal 8 Zeichen. An allen drei Stellen ist derselbe Wert zu verwenden. Ein `fsname` Schlüssel auf oberster Ebene hat keine Auswirkung.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
MGS NIDs für die Zielregistrierung	`eseries_lustre_filesystem_mdt.format_options.mgsnode`, `eseries_lustre_filesystem_ost.format_options.mgsnode`	"ha_cluster.yml#L98", "#L118"	Clusterweit	Unter MDT und OST verschachtelt `format_options`. Die Frontend-LNet-NIDs beider Knoten des HA-Paares sind einzubeziehen. Zuerst sind die NIDs des Knotens aufzulisten, der als bevorzugter MGS-Besitzer konfiguriert ist, gefolgt von den NIDs des Partnerknotens.
Pacemaker Cluster-Name	`eseries_lustre_pacemaker_cib_properties_overrides.cluster_properties.cluster-name`	"ha_cluster.yml#L228"	Clusterweit	Unter Pacemaker CIB-Eigenschaftsüberschreibungen verschachtelt. Eindeutiger Name für den HA Cluster.
BMC-Fencing-Adressen	`eseries_lustre_pacemaker_fencing_agents.fence_redfish.ip`	"ha_cluster.yml#L211", "#L214", "#L217", "#L220"	Für jeden Lustre Server wiederholen.	Redfish BMC IP-Adresse für jeden Serverknoten.
Empfänger von Warn-E-Mails	`eseries_lustre_alert_email_list`	"ha_cluster.yml#L233"	Clusterweit	Empfänger für HA-Ressourcen und Fehlerbenachrichtigungen.
Mail Domäne	`eseries_lustre_mail_service_overrides.conf.mydomain`	"ha_cluster.yml#L240"	Clusterweit	Postfix Domain, die zum Versenden von Benachrichtigungs-E-Mails verwendet wird.
NTP Zeitquellen	`eseries_lustre_time_sync_overrides.server_pools`	"ha_cluster.yml#L250"	Clusterweit	Zeitquellen für die Clusterknoten. Ein NTP-Pool oder ein oder mehrere einzelne NTP-Server sind anzugeben. Die erforderliche `pool` oder `server` Direktive sowie Optionen wie `iburst` sind einzuschließen.
PCI-zu-Name-udev-Schnittstellenzuordnung	`eseries_ip_udev_rules`	"ha_cluster.yml#L172"	Wiederholung pro HCA-Steckplatz-Layout	Jedem PCI-Steckplatz wird ein persistenter Schnittstellenname zugeordnet (Werte in Zeile 182). Erfassung mit `lspci -nnvmm`.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
Lokales Paket-Repository (optional)	`eseries_common_yum_repos`	"ha_cluster.yml#L256", "#L261"	Clusterweit	Für Air-Gap-Installationen den Block einkommentieren und das Repository `url` festlegen (L264).
Server-Management-IP	`ansible_host`	"host_vars/lustre_01.yml#L10"	Für jeden Lustre Server wiederholen.	Eine `host_vars/lustre_NN.yml` Datei pro Server.
NVMe-oF Backend-Schnittstellen	`eseries_nvme_roce_interfaces` (RoCE) / `eseries_nvme_ib_interfaces` (InfiniBand)	"RoCE host_vars/lustre_01.yml#L21", "InfiniBand host_vars/lustre_01.yml#L17"	Für jeden Lustre Server wiederholen.	Backend-Schnittstellenadressen, die für den NVMe-oF-Speicherverkehr zu den E-Series Arrays verwendet werden.
LNet Cluster-Schnittstellen	`eseries_roce_interfaces` (RoCE) / `eseries_ipoib_interfaces` (InfiniBand)	"RoCE host_vars/lustre_01.yml#L47", "InfiniBand host_vars/lustre_01.yml#L38"	Für jeden Lustre Server wiederholen.	Frontend LNet-Schnittstellenadressen (RoCE-Werte bei L56, InfiniBand-Werte bei L47).
Corosync Knoten-IPs	`eseries_lustre_corosync_node_ips`	"host_vars/lustre_01.yml#L58"	Für jeden Lustre Server wiederholen.	Alle Knoten-IPs im HA-Cluster auflisten (Werte in Zeile 62).
Lustre Zielservice-NIDs	`lustre_target_service_nodes`	"host_vars/lustre_01.yml#L64"	Für jeden Lustre Server wiederholen.	Frontend-LNet-NIDs (`@o2ib`beider Knoten im HA-Paar einbeziehen, damit die Zieldienste nach einem Failover weiterhin erreichbar bleiben (Werte in Zeile 67).
Array Management-IP	`ansible_host`	"host_vars/netapp_01.yml#L13"	Wiederholung pro E-Series Array	Management-IP eines Controllers; die API-URL wird daraus abgeleitet.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
Mitgliedschaft in einer Baustein-Gruppe	`mgs` / `mds_NN` / `oss_NN`	"lustre_inventory.yml"	Pro Baustein	Jeder Inventar-Hostname muss mit dem Basisnamen der zugehörigen `host_vars` Datei übereinstimmen. Beispielsweise verwendet Host `lustre_01` `host_vars/lustre_01.yml`, und `netapp_01` verwendet `host_vars/netapp_01.yml`. Die MGS-Gruppe ist nur im ersten Baustein enthalten. MDS-Gruppen sind im Basis Baustein und in jedem MDT+OST-Erweiterungsbaustein enthalten; in reinen OST-Erweiterungsbausteinen sind MDS-Gruppen nicht enthalten. Erweiterungsziele werden mit dem vorhandenen MGS unter Verwendung von `mgsnode` registriert.

Die Einstellungen sind für jeden Host und Baustein zu wiederholen. Eine `host_vars/lustre_NN.yml` Datei pro Lustre Server und eine `host_vars/netapp_NN.yml` Datei pro Array sind zu erstellen, und die Inventargruppen sind für jeden zusätzlichen Baustein zu wiederholen.

Die Deployment-Playbooks laden Array-, Pacemaker- und BMC-Zugangsdaten aus der gemeinsam genutzten `building_blocks/passwords.yml` Datei. Diese Datei ist auszufüllen und vor dem Deployment mit Ansible Vault zu verschlüsseln, oder die Werte werden zur Laufzeit sicher mit `--extra-vars` überschrieben. Zugangsdaten im Klartext dürfen nicht in die Quellcodeverwaltung übernommen werden.

Beispiele hierfür sind IP-Adressen, Hostnamen, Schnittstellennamen und Volumengrößen in den Bereitstellungsvorlagen. Alle Inventardateien der Umgebung sollten vor dem Ausführen eines Playbooks angepasst werden.

Nachdem Sie fertig sind

"Bereitstellung des Lustre HA Clusters und der Clients".

Bereitstellung des Lustre HA Clusters und der Clients

Das Serverbetriebssystem wird vorbereitet, NVIDIA DOCA-OFED installiert, der Lustre HA Cluster bereitgestellt und Lustre Clients mithilfe von NetApp Ansible Playbooks konfiguriert.

Bevor Sie beginnen, die in "Das Lustre Ansible-Inventar anpassen" beschriebene Inventarliste vervollständigen und sichern.

Bereitstellung des Lustre HA Clusters

Schritte

1. Vom `building_blocks/playbooks` Verzeichnis in Ihrer Arbeitskopie aus wird das Betriebssystem auf den Lustre-Servern vorbereitet. Ersetzen Sie `<inventory_path>` durch den Pfad zu Ihrem benutzerdefinierten Inventarverzeichnis:

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

2. NVIDIA DOCA-OFED wird auf jedem Lustre Server installiert. Die Kernelmodule werden für den Kernel erstellt, der im vorherigen Schritt installiert wurde. Das Verfahren für Ihren Kernel befindet sich in "NVIDIA DOCA-Host Installation und Upgrade". Das folgende Beispiel zeigt eine DOCA-OFED Installation auf RHEL oder Rocky Linux.

   a. Installieren Sie das DOCA Host-Repository-Paket und aktualisieren Sie den Paketcache. Ersetzen Sie `<doca_host_package>` durch das DOCA Host-Paket für Ihr Betriebssystem und Ihre Architektur:

```
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. Erstellt die DOCA-Kernelmodule für den laufenden Kernel. Das Skript gibt den Pfad des erstellten Pakets aus:

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

   c. Installieren Sie das vom Skript erstellte Kernelmodul-Repository-Paket und aktualisieren Sie anschließend den Paketcache. Ersetzen Sie `<doca_kernel_repo>` mit dem Pfad aus dem vorherigen Schritt:

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

   d. Installieren Sie die DOCA-OFED Benutzerbereichs-, Kernel- und NVMe-Pakete. Ersetzen Sie `<kernel_version>` durch die Versionsnummer des laufenden Kernels:

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

```
dnf install -y kmod-mlnx-nvme
```

3. Das Hauptbereitstellungs-Playbook ausführen:

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

Passen Sie `--forks` die Größe der Bereitstellung an, um zu steuern, wie viele Hosts Ansible parallel konfiguriert. Zum Beispiel `--forks 20` für einen größeren Cluster.

Das Playbook stellt E-Series Volume-Gruppen oder DDP-Pools und Volumes bereit, konfiguriert die NVMe-oF-Host-Konnektivität, formatiert und mountet die Lustre-Targets, konfiguriert LNet, wendet Leistungsoptimierungen an und konfiguriert die Pacemaker HA-Ressourcen.

Eine Bereitstellung mit einem Baustein bildet ein Zwei-Node-Pacemaker-Cluster. Knoten A erhält im Falle eines Failovers eine zusätzliche Stimme. Um in einem Zwei-Node-Cluster den Quorum zu erreichen, sollte die Konfiguration eines qdevice in Betracht gezogen werden. Enthält eine Bereitstellung mehrere Bausteine, trägt jeder Baustein ein Serverpaar mit zwei Knoten zum größeren Pacemaker-Cluster bei, bis zum dokumentierten Clusterlimit. Die Fencing- und Quorum-Konfiguration im "HA Administrationshandbuch" sollte überprüft werden, damit der Cluster Ziele im Falle eines Knotenausfalls sicher wiederherstellen kann.

Lustre Clients bereitstellen

Lustre Clients können auf jedem System bereitgestellt werden, das einen Dateisystemzugriff benötigt, indem die Client-Bereitstellungsvorlage angepasst und das Client-Playbook ausgeführt wird.

Schritte

1. Wählen Sie ein kompatibles Client-Betriebssystem und einen Kernel aus der "Lustre Support-Matrix" aus. Die Lustre Client-Pakete für diesen Kernel können von "netapp-lustre" erstellt und installiert werden, falls sie noch nicht installiert sind.
2. Erstellen Sie eine Arbeitskopie des `clients`Vorlagenverzeichnisses, einschließlich der gemeinsamen `passwords.yml`Datei, und wählen Sie die Vorlage aus, die zu Ihrem Fabric passt:
   - **InfiniBand:** "IB_lustre_cluster_clients"
   - **RoCE:** "ROCE_lustre_cluster_clients"
3. Passen Sie `lustre_client_inventory.yml`, `host_vars` und `group_vars` für Ihre Kunden an. Zeilenverweise in der Tabelle verweisen auf die RoCE Clientvorlagendateien im `ansible-lustre release/1.0.0`-Tag; die InfiniBand Clientvorlage verwendet dieselben Variablen, sofern nicht anders angegeben.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
Client Hosts	`lustre_clients`	"lustre_client_inventory.yml #L4"	Pro Client	Jeder Client-Hostname wird aufgelistet.

Feld	Ansible Variable	Vorlagendatei (`release/1.0.0`)	Umfang	Hinweise
Client-SSH-Benutzer und Passwort für „become“	`ssh_client_user` / `ssh_client_become_pass`	"passwords.yml#L2"	Clusterweit	Das Passwort sollte nur gesetzt werden, wenn der SSH-Benutzer nicht `root` ist.
LNet Schnittstellen	`eseries_lustre_lnet`	"group_vars/all.yml#L7"	Clusterweit	Namen der clientseitigen LNet-Schnittstellen (Werte bei L13).
MGS-NIDs einhängen	`lustre_client_mounts.mgsnode`	"group_vars/all.yml#L17"	Clusterweit	MGS NIDs, die zum Einbinden des Dateisystems verwendet werden (Werte in Zeile 20).
Dateisystemname	`lustre_client_mounts.fsname`	"group_vars/all.yml#L21"	Clusterweit	Muss mit dem Cluster übereinstimmen `fsname`.
Mount-Punkt	`lustre_client_mounts.mount_point`	"group_vars/all.yml#L22"	Clusterweit	Client Mount-Verzeichnis.
Client-Management-IP	`ansible_host`	"host_vars/client_01.yml#L4"	Wiederholung pro Kunde	Eine `host_vars/client_NN.yml` Datei pro Kunde.
Storage-Fabric-Schnittstellen	`eseries_roce_interfaces`	"host_vars/client_01.yml#L10"	Wiederholung pro Kunde	Frontend-Schnittstellenadressen (Werte in Zeile 15). Die InfiniBand-Vorlage verwendet `eseries_ipoib_interfaces` hier.

Die Einstellungen sind für jeden Client zu wiederholen. Für jeden Client ist eine `host_vars/client_NN.yml` Datei zu erstellen und der entsprechende Host zu `lustre_client_inventory.yml` hinzuzufügen. Die gemeinsame `clients/passwords.yml` Datei ist zu befüllen und vor dem Ausführen des Client Playbooks mit Ansible Vault zu verschlüsseln.

4. Das Client-Playbook aus dem Client-Vorlagenverzeichnis ausführen:

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

Das Client-Playbook konfiguriert die Client-Netzwerkschnittstellen und LNet und kann eine persistente systemd Mount-Unit für das Lustre Dateisystem erstellen.

Nachdem Sie fertig sind

"Die Bereitstellung überprüfen".

Lustre Bereitstellung überprüfen und erweitern

Vor dem Produktiveinsatz sind der Zustand des Clusters, die Lustre-Mounts und die Dateisystemkapazität zu bestätigen, anschließend kann dasselbe Inventar verwendet werden, um bei Bedarf Bausteine hinzuzufügen.

Die Bereitstellung überprüfen

Schritte

1. Von einem Lustre-Server aus lässt sich bestätigen, dass alle Pacemaker-Ressourcen auf den erwarteten Knoten gestartet sind:

```
pcs status
```

2. Von einem Lustre-Client aus lässt sich bestätigen, dass das Lustre-Dateisystem eingebunden ist:

```
mount -t lustre
```

3. Von einem Lustre-Client aus lässt sich die Dateisystemkapazität sowie die Sichtbarkeit der MDT- und OST-Ziele bestätigen:

```
lfs df -h
```

Zur Leistungsvalidierung mit IOR, mdtest und fio siehe "Größenberatung".

Für kontrollierte Zielmigrationen oder HA-Failover-Tests sind die Verfahren in der "HA Administrationshandbuch" zu befolgen.

Dateisystem erweitern

Um die Kapazität oder die Metadatenleistung zu erhöhen, wird das bestehende Inventar um einen zusätzlichen Baustein erweitert. Für den neuen Baustein wird kein separates Inventar erstellt. Das Playbook weist automatisch eindeutige MDT- und OST-Indizes zu und registriert die neuen Ziele beim bestehenden MGS.

Schritte

1. Fügen Sie die neuen Bausteinhosts, Arrays und Ressourcengruppen (MDT+OST oder nur OST) demselben Inventar sowie den `host_vars` und `group_vars` Dateien hinzu, die für die ursprüngliche Bereitstellung verwendet wurden.
2. Das Hauptbereitstellungs-Playbook wird erneut für das aktualisierte Inventar ausgeführt.

Siehe "Lösungsarchitektur"für Skalierungsregeln und "Größenberatung"für Hinweise, wann welcher Baustein hinzugefügt werden sollte.

Lustre mit NetApp E-Series Storage – Zugehörige Ressourcen

Verwenden Sie diese Links für zugehörige Dokumentation, Tools und Software, wenn eine Lustre mit NetApp E-Series Storage-Implementierung geplant oder erweitert wird. Informationen zur Dimensionierung und zu den Implementierungsschritten finden sich unter "Größenberatung" und "Die Lösung bereitstellen".

NetApp Ressourcen

- "NetApp Interoperabilitätsmatrix-Tool". Suche nach **E-Series Lustre**.
- "NetApp EF-Series All-Flash-Array Datenblatt"
- "NetApp SANtricity System Manager Dokumentation"
- "NetApp ansible-lustre Collection". Hinweise zur Bestandsaufnahme sind unter "Das Lustre Ansible-Inventar anpassen" zu finden.
- "NetApp Lustre-Quellcode-Repository". NetApp stellt vorkompilierte Lustre-Server-RPMs für Rocky Linux 9.8 und Red Hat Enterprise Linux 9.8 bereit. Lustre-Clientpakete für das Client-Betriebssystem und den Kernel werden aus dem NetApp-Quellcode erstellt.

Lustre Community

- "Lustre Dateisystem"
- "Lustre Support-Matrix" für Client-Betriebssysteme und Kernel
- "Lustre Wiki: Lustre Tuning"

Verwandte NetApp KI-Infrastrukturlösungen

- "BeeGFS auf NetApp mit E-Series Storage"
- "Bereitstellung von IBM Spectrum Scale mit NetApp E-Series Storage"
- "NVIDIA DGX SuperPOD mit NetApp EF-Serie"

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGENDEINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter http://www.netapp.com/TM aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.