



Lustre con NetApp storage E-Series

NetApp artificial intelligence solutions

NetApp
August 31, 2026

Sommario

Lustre con NetApp storage E-Series	1
Lustre con NetApp E-Series Storage - Introduzione	1
Panoramica della soluzione	1
Lustre con storage NetApp E-Series - Architettura della soluzione	1
Design del building block	2
Raccomandazioni di scaling	3
Architettura HA	4
Architettura di rete	5
Client Lustre	5
Lustre con NetApp E-Series Storage - Componenti hardware	6
Requisiti del server	6
building block standard EF80	8
Selezione del pool di storage: TLC vs QLC	12
Requisiti di rete	12
Lustre con NetApp E-Series Storage - Componenti software	12
Software building block	12
Automazione Ansible	13
Software client Lustre	14
Lustre con NetApp E-Series Storage - Guida al dimensionamento	14
Dimensionamento della capacità	14
Scalabilità	15
Prestazione	16
Distribuisci la soluzione	16
Implementa Lustre con NetApp E-Series Storage	16
Hardware Lustre per rack e cavi	17
Prepara l'ambiente di deployment di Lustre	19
Personalizza l'inventario Lustre Ansible	20
Distribuisci il cluster Lustre HA e i client	23
Verifica ed espandi la distribuzione di Lustre	26
Lustre con NetApp E-Series Storage - Risorse correlate	27
Risorse NetApp	27
Community Lustre	28
Soluzioni di infrastruttura AI correlate NetApp	28

Lustre con NetApp storage E-Series

Lustre con NetApp E-Series Storage - Introduzione

Lustre con NetApp E-Series Storage è una soluzione di file system parallelo validata per carichi di lavoro AI e HPC, costruita a partire da building block ripetibili composti da coppie di server HA e array all-flash E-Series.

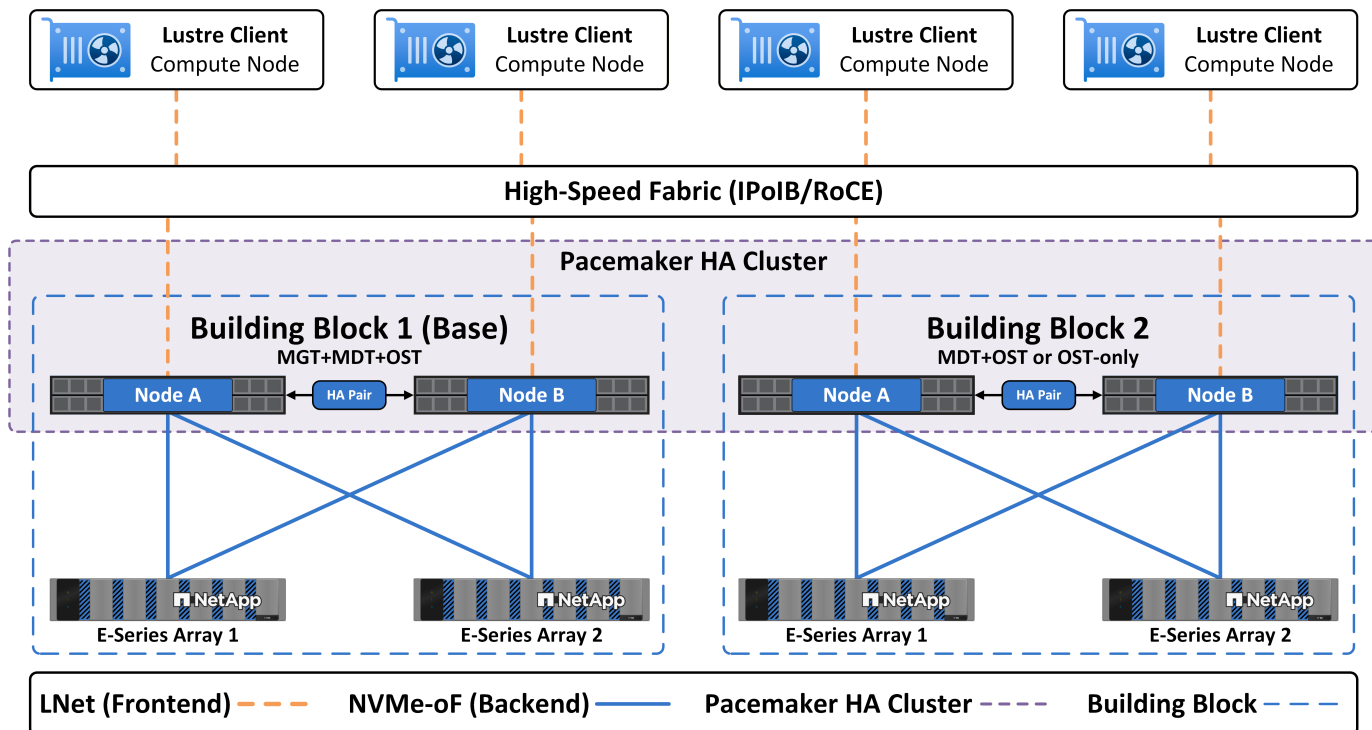
Panoramica della soluzione

Lustre con NetApp E-Series Storage combina il file system parallelo open-source Lustre con lo storage a blocchi NetApp E-Series per offrire una base scalabile e dalle performance elevate per carichi di lavoro ad alta intensità di dati. Ogni building block abbina due nodi server OSS/MDS configurati per l'alta disponibilità (HA) con due array all-flash E-Series. La connettività NVMe-oF di backend trasporta l'I/O a blocchi verso gli array e una rete LNet separata collega i client e i nodi server OSS/MDS per l'accesso parallelo al file system.

Un building block di base ospita il server di gestione (MGS) e i target di metadati iniziali (MDT) e i target di storage a oggetti (OST). Ulteriori building block si registrano rispetto all'MGS di base per scalare la capacità dei metadati, la capacità dei dati e il throughput man mano che aumentano le esigenze di carico di lavoro.

La figura seguente mostra un esempio di implementazione con più building block e client Lustre connessi tramite LNet.

Panoramica della soluzione di storage Lustre con NetApp E-Series



Lustre con storage NetApp E-Series - Architettura della soluzione

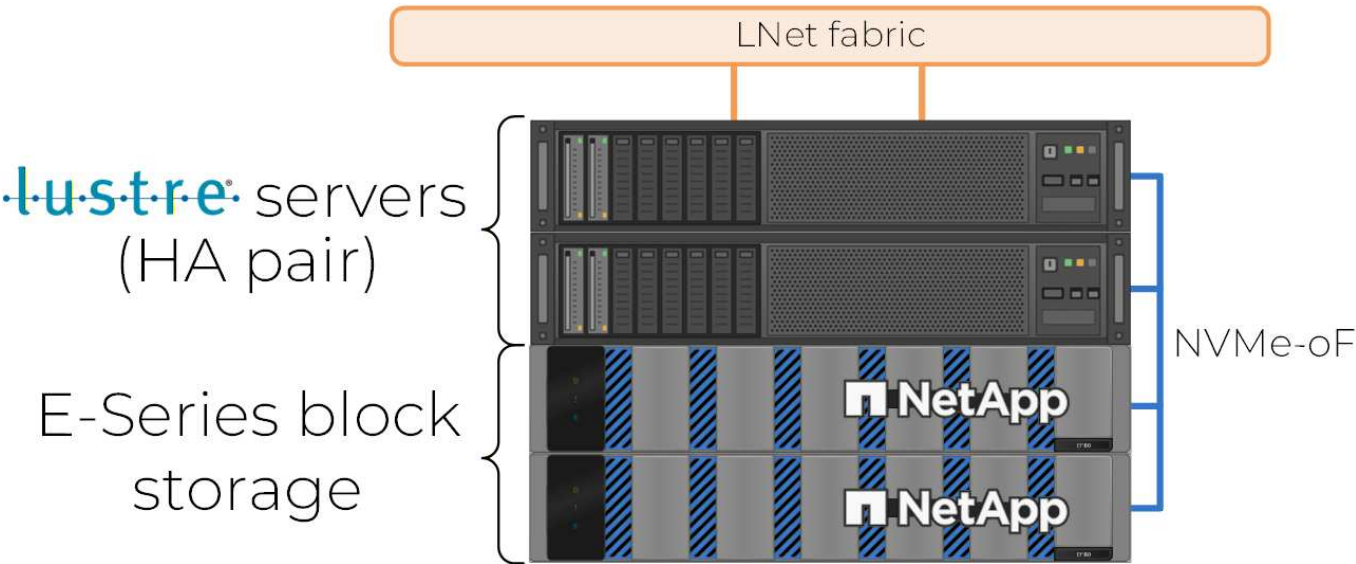
Scopri come Lustre con NetApp E-Series storage utilizza building block, alta disponibilità,

topologia di rete e client così puoi pianificare capacità, prestazioni e failover.

Design del building block

Un building block è l'unità scalabile fondamentale della soluzione NetApp E-Series Lustre. Ogni building block è costituito da due nodi server configurati come una coppia ad alta disponibilità (HA) che forniscono le funzioni di Lustre object storage server (OSS) e metadata server (MDS), due NetApp E-Series all-flash array, connettività backend NVMe over Fabrics (NVMe-oF) dedicata tra server e array, e connettività frontend di rete Lustre (LNet) dedicata per i client e la comunicazione tra nodi. Un cluster HA Pacemaker e Corosync può contenere la coppia di server di uno o più building block. La raccolta NetApp `ansible-lustre` utilizza l'automazione Ansible per distribuire e configurare i servizi Lustre.

building block Lustre



I building block si adattano ai requisiti di crescita del file system. Ogni file system richiede un building block di base. Il building block di base ospita il management server (MGS) su un management target (MGT) e fornisce la capacità iniziale di metadata target (MDT) e object storage target (OST). I building block aggiuntivi estendono il file system e registrano i loro target MDT e OST sull'MGS del building block di base. L'approccio modulare consente di scalare capacità e prestazioni in modo indipendente in base alle esigenze del carico di lavoro.

NetApp consiglia i seguenti profili di configurazione dei building block per la maggior parte delle implementazioni. I conteggi di MGS, MDT e OST elencati sono valori predefiniti consigliati, ottimizzati per massimizzare le prestazioni e la capacità degli storage E-Series. Regola il numero di target, le dimensioni dei volumi e l'allocazione dei drive E-Series nell'inventario Ansible in base alle esigenze del carico di lavoro. Ad esempio, le implementazioni che richiedono una maggiore capacità di storage dei metadati possono prevedere più MDT e meno OST all'interno dello stesso hardware building block.

La tabella seguente riassume i tipi di building block e le quantità target consigliate.

Tipo	MGS	MDTs	OST	Usa
Base	1	8	32	Primo building block di un file system. Ospita l'MGS e la capacità iniziale di MDT e OST.

Tipo	MGS	MDTs	OST	Usa
MDT+OST	0	8	32	Aggiunge metadati e capacità dati. Registra rispetto al building block di base MGS.
Solo OST	0	0	32	Aggiunge solo capacità dati. Registrati rispetto al building block MGS.

- **Tipo di unità e configurazione del pool:** E-Series supporta i tradizionali gruppi di volumi RAID e i Dynamic Disk Pools (DDP). Per le unità NVMe TLC, usa gruppi di volumi RAID 6 per lo storage OST e gruppi di volumi RAID 1 per lo storage MGS e MDT. Per le unità NVMe QLC, usa DDP per il pool di unità condiviso.
- **Distribuzione dei target:** I target Lustre in ogni building block sono bilanciati tra entrambi i nodi server OSS/MDS e entrambi gli array E-Series. Il layout distribuisce l'I/O tra le CPU dei server e i controller degli array per migliorare le prestazioni del percorso NVMe-oF e mantenere la ridondanza. Vedi ["distribuzione dei target e connettività NVMe-oF"](#) in ["Componenti hardware"](#).

I sistemi operativi supportati, le versioni di Lustre, il firmware dell'array e i relativi building block sono elencati in ["NetApp Interoperability Matrix Tool \(IMT\)"](#) sotto **E-Series Lustre**. Per dettagli sul numero di volumi, la disposizione dei dischi e le linee guida per il dimensionamento, vedi ["Componenti hardware"](#). Per i componenti software, vedi ["Componenti software"](#).

Raccomandazioni di scaling

Il file system Lustre si adatta aggiungendo building block quando la capacità dei metadati, la capacità dei dati o entrambe diventano limitanti. Scala gli MDT in base al numero di file e alle IOPS dei metadati; scala gli OST in base alla capacità e ai requisiti aggregati di throughput.

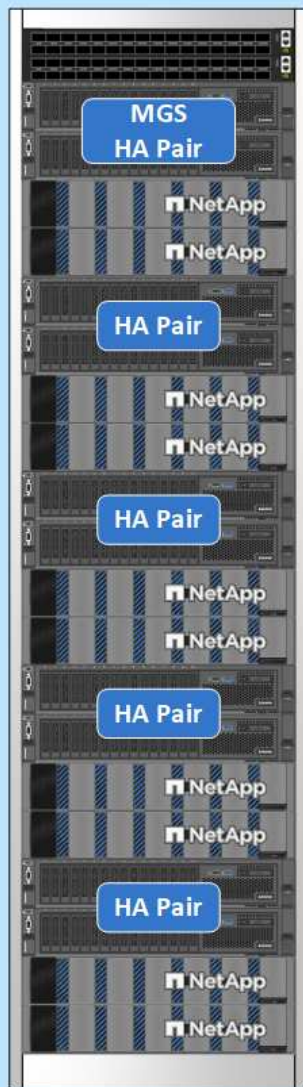
1. **Un building block per ogni file system:** Distribuisci esattamente un building block; esso ospita l'MGS e i target MDT e OST iniziali.
2. **Profilo building block aggiuntivo:** Aggiungi un building block solo OST quando la capacità MDT esistente è sufficiente e la capacità dati o il throughput devono aumentare. Aggiungi un building block MDT+OST quando le prestazioni dei metadati o la capacità dello spazio dei nomi diventano il collo di bottiglia.
3. **Limite del cluster HA:** Limita ciascun cluster Pacemaker e Corosync HA a cinque building block (dieci nodi server Lustre). Suddividi equamente le implementazioni più grandi in più cluster HA per evitare limitazioni di risorse in configurazioni di grandi dimensioni.

La figura seguente mostra fino a cinque building block impilati in un rack da 42U (un cluster Pacemaker HA per rack). Più rack possono partecipare a un singolo file system Lustre; solo il building block di base ospita l'MGS.

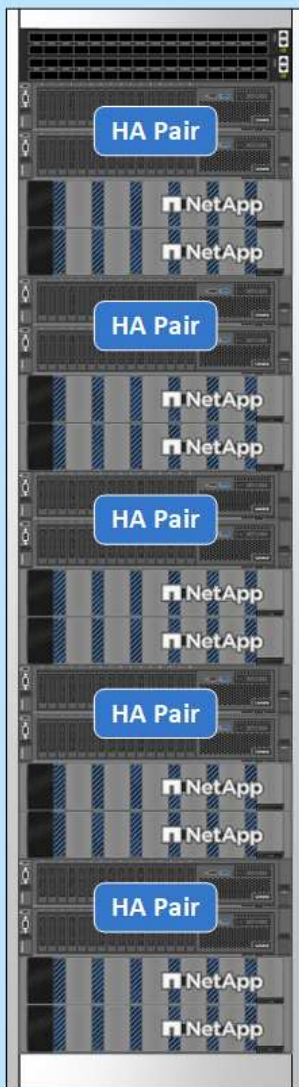
Scalabilità rack del file system parallelo Lustre

Lustre Parallel Filesystem

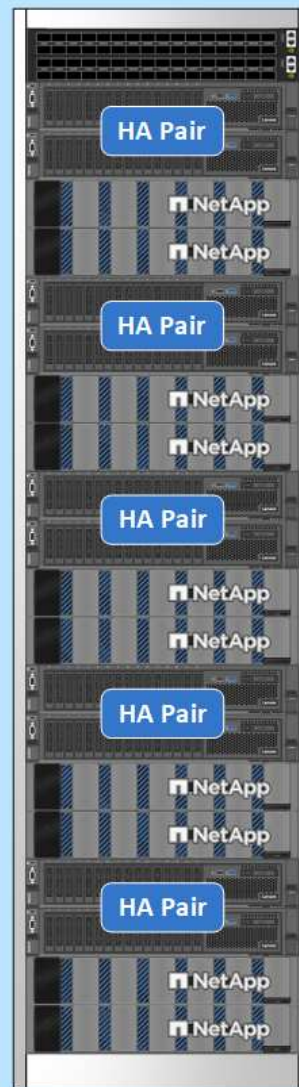
Standalone HA Cluster



Standalone HA Cluster



Standalone HA Cluster



Per esempi di dimensionamento, vedi ["Guida alle taglie"](#).

Architettura HA

Lustre con NetApp E-Series storage utilizza un'architettura ad alta disponibilità (HA) con dischi condivisi integrata con NetApp E-Series storage. Pacemaker gestisce le risorse HA e Corosync fornisce appartenenza al cluster e messaggistica. Insieme, gestiscono il failover del target Lustre tra i due nodi server OSS/MDS in ciascun building block. Multipath NVMe-oF collega entrambi i nodi server OSS/MDS agli stessi volumi E-Series, così che ciascun nodo possa assumere il controllo dei servizi target quando necessario.

Ogni MGS, MDT e OST è configurato come risorsa HA con le sue dipendenze in un gruppo di risorse Pacemaker. Pacemaker garantisce che le risorse si avviino e si arrestino nell'ordine corretto, rimangano collocate sullo stesso nodo e vengano eseguite su un solo nodo alla volta. L'accesso simultaneo alla stessa destinazione da entrambi i nodi potrebbe danneggiare il file system.

- **Failover del target:** Entrambi i nodi server OSS/MDS sono peer nel cluster, ma ogni target Lustre è attivo su un solo nodo alla volta. Una risorsa di monitoraggio Pacemaker controlla lo stato di salute di ciascun target e delle sue dipendenze e attiva un failover quando un target diventa non disponibile sul nodo corrente. In caso di errore, Pacemaker riavvia il gruppo di risorse interessato sul nodo partner nell'ordine corretto e i client si riconnettono in modo trasparente al target nella sua nuova posizione una volta che i servizi riprendono. Poiché ogni target si attiva su un singolo nodo, il file system non è mai esposto a scritture simultanee da entrambi i nodi.
- **Accesso allo storage condiviso:** I percorsi NVMe-oF multipath collegano ciascun nodo server OSS/MDS a tutti i controller E-Series nel building block, così ogni volume di destinazione è raggiungibile da entrambi i nodi. Se un nodo server si guasta, il nodo partner ha già percorsi attivi verso gli stessi volumi e può assumere la proprietà delle destinazioni senza dover riconfigurare la connettività di storage. Se un controller dell'array o un percorso si guasta, il multipath reindirizza l'I/O a un controller funzionante. Questa doppia ridondanza permette ai target Lustre di effettuare il failover tra i nodi server OSS/MDS o i controller dell'array senza perdere l'accesso ai volumi di supporto.
- **STONITH fencing:** Quando si verifica un errore, Pacemaker a volte non riesce a comunicare con il nodo difettoso per confermare che i suoi target siano stati arrestati. Prima di riavviare tali target altrove, Pacemaker isola il nodo difettoso, idealmente interrompendo l'alimentazione, per assicurarsi che sia spento. Il fencing impedisce uno scenario di split-brain in cui entrambi i nodi accedono contemporaneamente allo stesso target e corrompono il file system sottostante, preservando l'integrità dei dati durante il failover. NetApp consiglia `fence_redfish` per server con controller di gestione della scheda madre (BMC) compatibili con Redfish. Sono supportati anche altri agenti di fencing, come `fence_apc`.

Per l'amministrazione del cluster e la configurazione del fencing, vedi il ["Guida utente HA"](#) nel ["collezione ansible-lustre"](#).

Architettura di rete

La soluzione utilizza percorsi di rete separati per il traffico di backend dello storage e per il traffico di frontend LNet.

- **Backend (NVMe-oF):** I nodi server OSS/MDS si connettono agli array E-Series tramite NVMe-oF utilizzando NVMe/InfiniBand o NVMe/RoCE. Il percorso NVMe-oF trasporta l'I/O a blocchi tra i servizi target Lustre e i volumi E-Series. Per il cablaggio backend EF80 six-HCA, vedi ["Hardware Lustre rack e cavi"](#). Per il posizionamento preferito dei target tra nodi e array, vedi ["distribuzione target"](#) in ["Componenti hardware"](#).
- **Frontend (LNet):** I client Lustre e i nodi server OSS/MDS comunicano tramite LNet utilizzando il `@o2ib` tipo di rete con lo stack NVIDIA OFED. InfiniBand e RoCE sono entrambi trasporti frontend validati. La raccolta `ansible-lustre` configura tutte le interfacce frontend per LNet multi-rail, che aggrega più porte per aumentare la larghezza di banda e fornire ridondanza di percorso per il traffico client e tra nodi.
- **Gestione e cluster:** Una rete di out-of-band management fornisce la gestione dei server, l'accesso BMC e la gestione degli array. Gli agenti di fencing STONITH, come `fence_redfish`, utilizzano questa rete per raggiungere i BMC dei server e togliere alimentazione a un nodo guasto. Corosync può anche eseguire la comunicazione del cluster su questa rete come anello aggiuntivo insieme al fabric frontend. Configurare Corosync con più anelli aggiunge ridondanza per la messaggistica del cluster e riduce il rischio che un singolo guasto di rete interrompa il quorum.

Client Lustre

Un client Lustre carica il modulo del kernel client, stabilisce la connettività LNet ai nodi server OSS/MDS e monta il file system per presentare alle applicazioni un singolo namespace coerente e conforme a POSIX. L'I/O è distribuito in parallelo tra gli OST attivi. I nodi client sono esterni a un building block e utilizzano il file

system tramite il fabric frontend LNet. I sistemi operativi client non sono elencati in IMT per questa soluzione; usa la ["Matrice di supporto Lustre"](#) per le distribuzioni client testate e le versioni del kernel supportate dalla release di Lustre distribuita (ad esempio, Red Hat Enterprise Linux 9, SUSE Linux Enterprise Server 15 e Ubuntu 24.04).

I nodi client richiedono la connettività alla stessa LNet fabric e allo stesso tipo di rete dei nodi server OSS/MDS. Se necessario, un router LNet può collegare subnet o fabric LNet separate.

NetApp fornisce pacchetti RPM precompilati per il server Lustre per Rocky Linux 9.8 e Red Hat Enterprise Linux 9.8. NetApp non fornisce pacchetti client precompilati. Compila i pacchetti client per il sistema operativo e il kernel client dal codice sorgente di Lustre in ["repository netapp-lustre"](#).

Dopo l'installazione dei pacchetti client, il ruolo opzionale `lustre_client` nella ["collezione ansible-lustre"](#) configura le interfacce di rete client e LNet, abilita LNet multi-rail quando selezionato, convalida la connettività e gestisce un'unità di mount systemd persistente. Il ruolo non compila Lustre. Installa i pacchetti client solo quando `lustre_client_packages` è popolato. Se vuoi, puoi configurare manualmente i client. Per le procedure dettagliate, vedi ["Distribuisci la soluzione"](#) e ["documentazione del ruolo lustre_client"](#).

Lustre con NetApp E-Series Storage - Componenti hardware

Usa questi requisiti hardware per server, array NetApp E-Series, layout dello storage e networking quando dimensioni e ordini Lustre con NetApp E-Series Storage. Per i componenti software, vedi ["Componenti software"](#).

Requisiti del server

La soluzione utilizza un modello bring-your-own-server (BYOS). Ogni building block richiede due nodi server OSS/MDS che soddisfano o superano le seguenti specifiche.

Specifiche del nodo

La tabella seguente elenca i requisiti minimi del server per ciascun nodo OSS/MDS.

Componente	Requisito
Quantità	2 nodi per building block
CPU	AMD EPYC o Intel Xeon, 32 core o superiore
Memoria	256 GB DDR5 (minimo)
HCA di rete	6 HCA a doppia porta da 200 Gb (vedi Connettività HCA)
Slot PCIe	2× PCIe Gen5 x16 e 4× PCIe Gen5 x8 (vedi Requisiti dello slot PCIe)
Unità di avvio	2× unità in RAID 1 (si consiglia RAID software o hardware)

NetApp ha validato la soluzione utilizzando server Lenovo ThinkSystem SR665 V3. Server equivalenti di altri produttori sono supportati se soddisfano questi requisiti.

Connettività HCA

La tabella seguente elenca i requisiti HCA, la porta frontend LNet e i requisiti del percorso NVMe-oF per ciascun nodo server OSS/MDS.

HCA per nodo	Porte LNet per nodo Lustre	Percorsi NVMe-oF per nodo Lustre	Esempio HCA
6 (12 porte)	4	8 in totale (4 per ogni array)	MCX755106AS-HEAT (scheda PCIe gen5 dual port 200Gb)

NetApp consiglia sei HCA per nodo per massimizzare la larghezza di banda degli array di storage EF80.

Requisiti dello slot PCIe

Ogni nodo server OSS/MDS in un building block EF80 contiene sei HCA a doppia porta: due per LNet e quattro per NVMe-oF. La larghezza dello slot determina la larghezza di banda disponibile per ogni HCA, quindi controlla la larghezza elettrica di ogni slot e riser invece che solo la larghezza della scheda. Una scheda Gen5 x16 installata in uno slot Gen5 x8 funziona a x8.

- **LNet HCA:** Installa negli slot PCIe Gen5 x16 per raggiungere il massimo throughput.
- **Schede HCA NVMe-oF:** Installa negli slot PCIe Gen5 x8 o PCIe Gen5 x16.
- **Bilanciamento NUMA:** Dividi gli HCA in modo uniforme tra le due zone NUMA così che ciascuna zona ospiti due interfacce LNet e quattro interfacce NVMe-oF.

La tabella seguente elenca il numero minimo di slot per ciascun nodo server OSS/MDS in un building block EF80.

tipo di traffico	HCA per nodo	Larghezza minima dello slot	Slot per zona NUMA
LNet	2	PCIe Gen5 x16	1
NVMe-oF	4	PCIe Gen5 x8	2

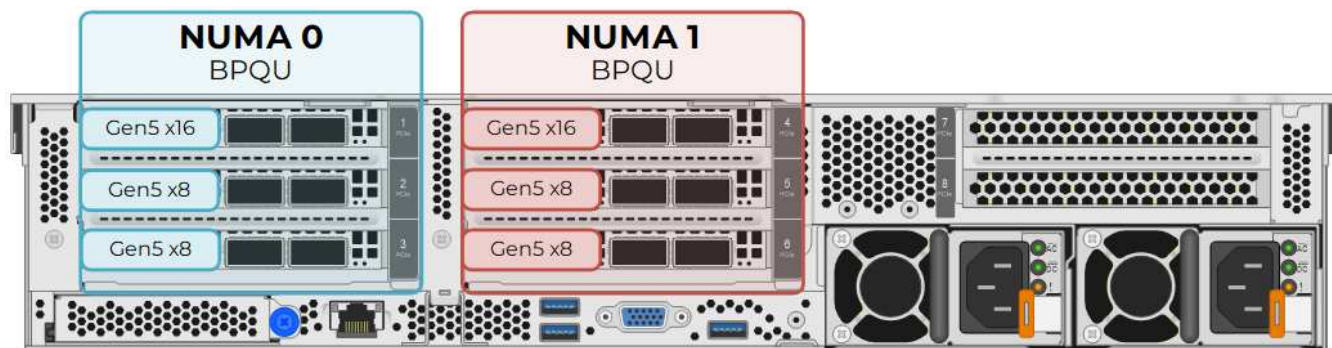
Facoltativamente, puoi suddividere le porte su un HCA a doppia porta in modo che una porta gestisca il traffico LNet e l'altra il traffico NVMe-oF. Installa tutti e quattro questi HCA in slot PCIe Gen5 x16 così che ogni porta LNet raggiunga il throughput massimo, poi aggiungi altri due HCA in slot Gen5 x8 o Gen5 x16 per le restanti porte NVMe-oF.

▼ Esempi di configurazioni riser per Lenovo ThinkSystem SR665 V3

Entrambe le seguenti combinazioni di montanti soddisfano i requisiti di slot per un building block EF80.

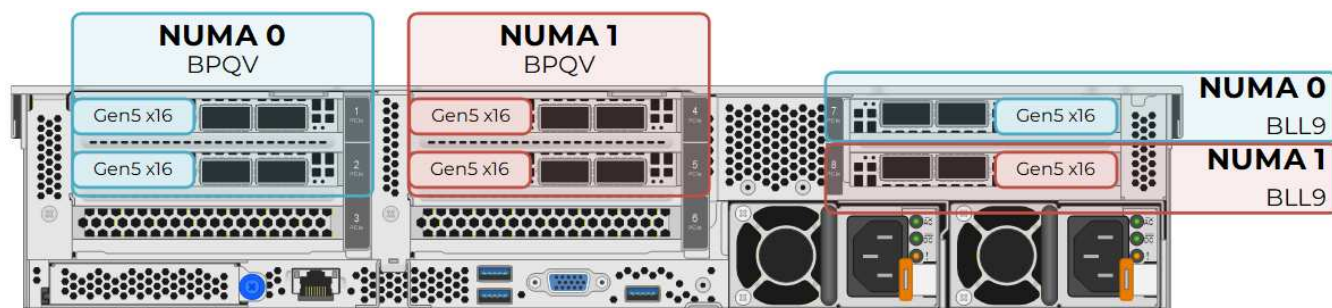
Larghezze minime degli slot

Installa un riser BPQU nelle posizioni riser 1 e 2. Ogni riser BPQU fornisce uno slot PCIe Gen5 x16 e due slot PCIe Gen5 x8. Insieme, i riser forniscono due slot Gen5 x16 per LNet e quattro slot Gen5 x8 per NVMe-oF, equamente suddivisi tra le zone NUMA.



Tutti gli slot Gen5 x16

Installa un riser BPQV nelle posizioni 1 e 2 e un riser BLL9 nella posizione 3. Ogni riser BPQV fornisce due slot PCIe Gen5 x16. Il riser BLL9 fornisce lo slot 7 nella zona NUMA 0 e lo slot 8 nella zona NUMA 1, entrambi PCIe Gen5 x16. Questa combinazione fornisce sei slot Gen5 x16, tre per zona NUMA, e supporta la suddivisione del traffico LNet e NVMe-oF tra le porte dello stesso HCA.



building block standard EF80

L'EF80 è la piattaforma standard convalidata per questa release della soluzione.

Specifiche dell'array

La tabella seguente elenca le specifiche dell'array EF80 per un building block (due array).

Componente	Specifiche
Modello	NetApp EF80
Fattore di forma	chassis base 2U, 24 slot SSD NVMe interni
Controllori	Doppio controller (A e B)
Dischi	24× SSD NVMe per array
Connettività I/O	8 porte host NVMe/IB o NVMe/RoCE da 200 Gb per array nel design Lustre convalidato

La piattaforma EF80 supporta fino a dodici porte host per array quando in ogni controller sono installati tre moduli I/O host a due porte. Il design Lustre validato utilizza moduli I/O host negli slot 1 e 2 per otto porte host per array.

Ogni array EF80 utilizza anche il modulo I/O dedicato dello slot 4 per il mirroring tra controller. Collega la porta 4a del controller A alla porta 4a del controller B e la porta 4b del controller A alla porta 4b del controller B. Queste connessioni forniscono il mirroring della cache e il trasferimento I/O e non vengono utilizzate per il

traffico NVMe-oF dell'host. Vedi ["Collega i cavi di mirroring tra i controller EF50 ed EF80"](#).

Per le specifiche complete di EF-Series per tutti i modelli, consulta il ["NetApp EF-Series array all-flash datasheet"](#).

Layout dei drive (24 drive per array)

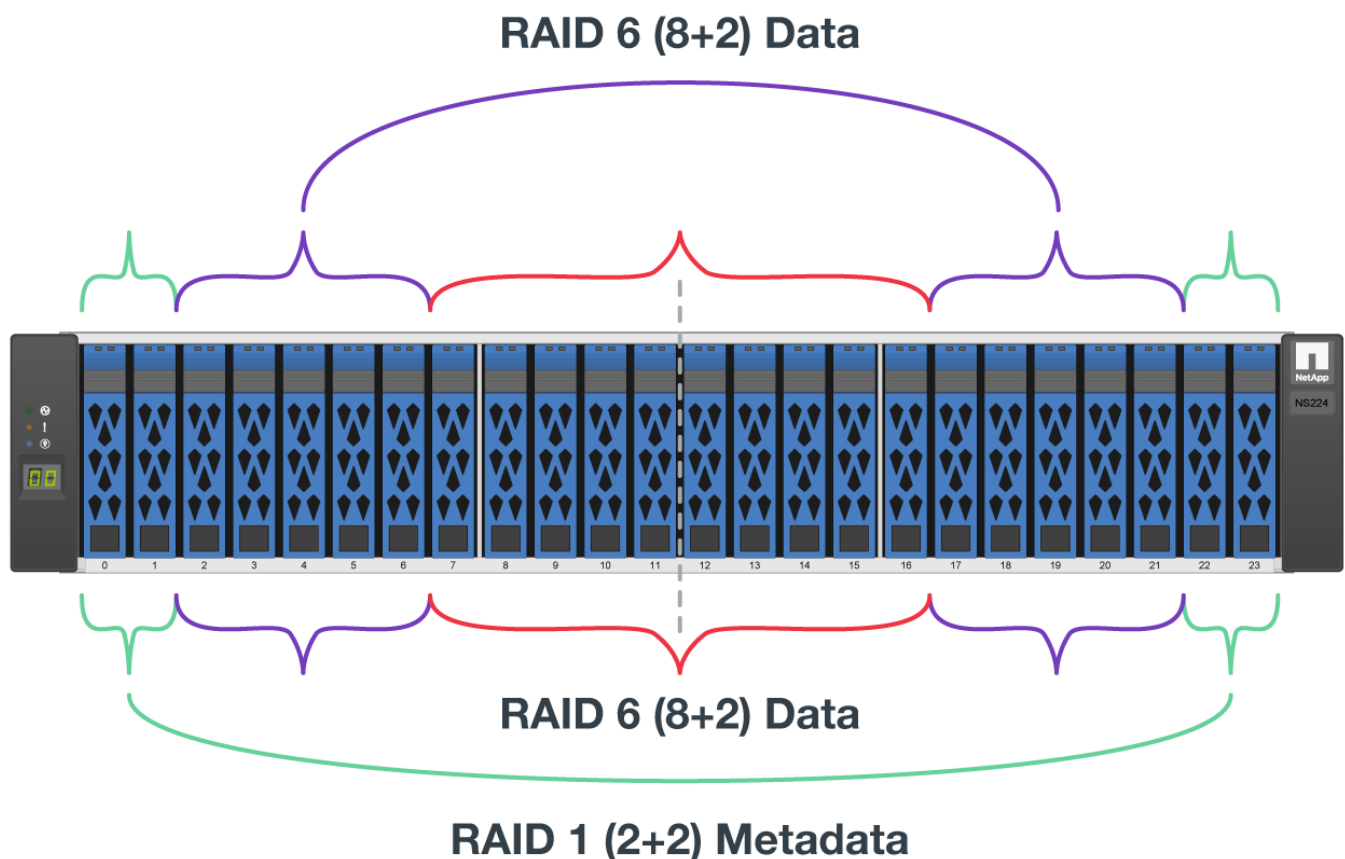
Ogni array EF80 in un building block utilizza tutti i ventiquattro slot per unità NVMe:

- 4 unità in RAID 1 per l'archiviazione MGS/MDT (gruppo di volumi condivisi o allocazione DDP)
- 10 unità in RAID 6 per l'archiviazione OST (primo pool OST)
- 10 unità in RAID 6 per l'archiviazione OST (secondo pool OST)

Questa configurazione si applica quando usi unità da 3,84 TB, 7,68 TB o 15,3 TB con gruppi di volumi tradizionali. Quando usi unità Capacity Flash (QLC) da 30,7 TB o 61,4 TB, crea un singolo Dynamic Disk Pool su tutte e ventiquattro le unità e crea volumi RAID 1 all'interno del pool per i volumi MGS e MDT, mentre usi i volumi RAID 6 predefiniti per gli OST.



Le unità da 1,92 TB non sono attualmente consigliate per questa soluzione. Usa una delle capacità di unità validate elencate sopra.



Conteggi di volume e target (building block di base)

La tabella seguente elenca i conteggi MGS, MDT e OST per un building block di base.

Tipo di destinazione	Conteggio per BB	RAID / pool	Note
MGS	1	RAID 1	Solo building block di base; array 1
MDT	8	RAID 1	4 per array
OST	32	RAID 6 (TLC) o DDP (QLC)	16 per array

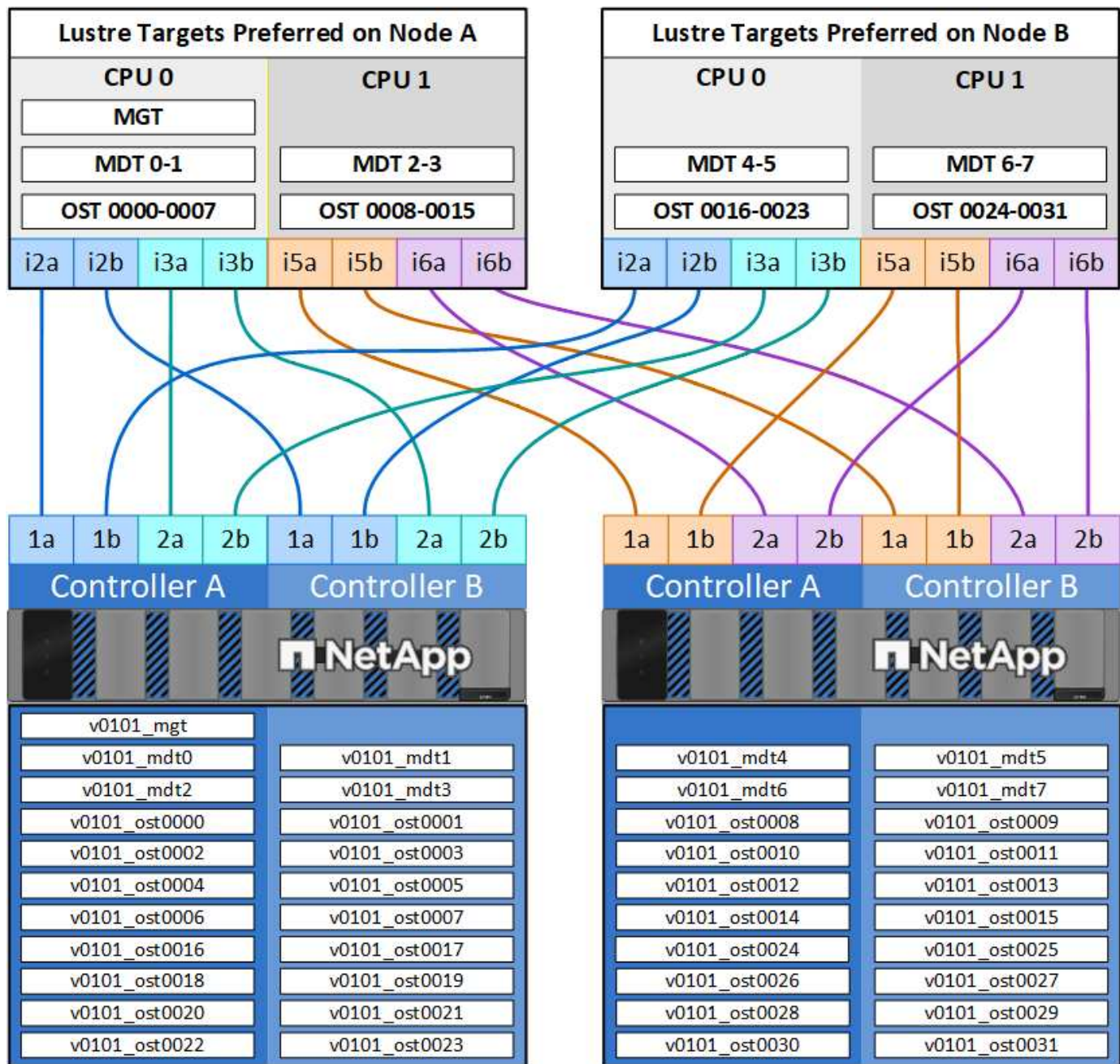
Utilizza le seguenti linee guida per il dimensionamento dei volumi come punto di partenza nell'inventario Ansible.

Volume	Taglia consigliata	Note
MGS	5–10 GiB	Solo dati di configurazione
MDT	Capacità RAID 1 ÷ conteggio MDT	Circa 1–2 TiB ciascuno (tipico)
OST	Capacità RAID 6 o DDP ÷ numero di OST per pool	Scala con la capacità del disco; vedi "Guida alle taglie"

Distribuzione del volume del building block primario

La figura seguente mostra il posizionamento preferito dei target Lustre e la connettività NVMe-oF tra i nodi server OSS/MDS e gli array E-Series in un building block EF80 di base. Il diagramma etichetta il management target come **MGT** (management target); un MGT ospita il servizio MGS (management server) a cui si fa riferimento altrove in questo documento.

Distribuzione dei target dei building block di base (EF80)



La tabella seguente elenca come i volumi sono distribuiti tra i due array e quale server preferisce ciascuna destinazione.

Tipo di destinazione	Array 1	Array 2	Server 1	Server 2	Note
MGS	1	0	1	0	Solo building block di base
MDT	4	4	4	4	Ogni server preferisce 2 MDT da ogni array
OST	16	16	16	16	8 per pool RAID 6 × 2 pool per array, o allocazione DDP equivalente
Volumi totali	21	20	21	20	

Selezione del pool di storage: TLC vs QLC

Scegli il tipo di pool E-Series in base alla capacità delle unità NVMe presenti negli array:

- **Unità da 3,84 TB, 7,68 TB e 15,3 TB:** Usa **gruppi di volumi RAID 6** per i volumi OST e **gruppi di volumi RAID 1** per i volumi MGS e MDT, oppure usa DDP per il pool di unità condivise.
- **Unità flash da 30,7 TB e 61,4 TB di capacità (QLC):** Usa solo **Dynamic Disk Pools (DDP)**. Crea volumi RAID 1 all'interno del DDP per lo storage MGS e MDT. Crea volumi RAID 6 per lo storage OST dal DDP.

Per le stime della capacità utilizzabile per building block in base alle dimensioni e alla disposizione dei drive, vedi ["Guida alle taglie"](#).

Requisiti di rete

Backend (NVMe-oF)

- NVMe/InfiniBand o NVMe/RoCE tra ciascun nodo OSS/MDS e ciascun array E-Series
- MTU 9000 su interfacce backend NVMe/RoCE (tipico)
- Otto percorsi da ciascun nodo Lustre verso gli array di storage (quattro per ciascun array)
- EF80 utilizza sei HCA per nodo (i2, i3, i5 e i6 per NVMe-oF; i1 e i4 per LNet). Collega il nodo A alle porte del controller le cui etichette terminano con a, come 1a e 2a. Collega il nodo B alle porte del controller le cui etichette terminano con b, come 1b e 2b. Vedi ["Cablaggio backend EF80 a sei HCA"](#).

Frontend (LNet)

- IPoB o RoCE per il tipo di rete (@o2ib LNet)
- MTU 9000 sulle interfacce frontend RoCE (tipico)
- Si consiglia la configurazione multi-rail LNet quando sono disponibili quattro porte frontend (EF80: i1 e i4 HCA, entrambe le porte)
- Rete di commutazione dedicata InfiniBand o RoCE che collega i nodi server OSS/MDS e i client Lustre
- Si consiglia RoCE lossless (PFC) su fabric RoCE.

Gestione

- Rete di gestione fuori banda per BMC dei server e porte di gestione degli array
- Rete Corosync dedicata o infrastruttura frontend condivisa (a seconda del sito)

Lustre con NetApp E-Series Storage - Componenti software

Esamina lo stack software e l'automazione Ansible che distribuisce Lustre con NetApp E-Series Storage. Per server, storage e reti, vedi ["Componenti hardware"](#).

Software building block

La tabella seguente elenca i componenti software utilizzati dagli array E-Series e dai nodi server OSS/MDS. Usa la ["NetApp Interoperability Matrix Tool \(IMT\)"](#) come fonte autorevole per le versioni e le combinazioni di componenti supportate. Cerca **E-Series Lustre** e verifica il sistema operativo, Lustre, SANtricity OS, DOCA-OFED, il firmware HCA e le versioni del pacchetto HA prima della distribuzione.

Componente	Funzione
OS SANtricity	Funziona su ogni array E-Series e offre SANtricity System Manager, alta disponibilità con controller dual-active e failover automatico del percorso I/O, bilanciamento del carico automatico e monitoraggio del percorso, Dynamic Disk Pools e RAID tradizionale, ricostruzione automatica delle unità, protezione della cache con mirroring, Data Assurance, monitoraggio proattivo dello stato di salute delle unità e modifiche online a software, firmware e configurazione.
Lustre	Offre ai client un singolo namespace conforme a POSIX, distribuendo metadati e dati dei file su più destinazioni per l'accesso parallelo. Il server di gestione (MGS) memorizza la configurazione del file system, i metadati target (MDT) gestiscono il namespace e gli object storage target (OST) memorizzano i dati dei file sui volumi E-Series. Questa separazione permette di scalare prestazioni e capacità aggiungendo building block.
Red Hat Enterprise Linux (RHEL) o Rocky Linux	Fornisce il kernel, i servizi di sistema e il framework di pacchetti per i servizi target di Lustre, il clustering HA e la connettività di storage e di rete. I repository del sistema operativo supportati forniscono anche Pacemaker, Corosync, fence agent, NVMe-oF e pacchetti multipath che vengono testati insieme come una soluzione stack.
Servizi cluster ad alta disponibilità (Pacemaker, Corosync e fence agent)	Insieme, questi servizi garantiscono un'elevata disponibilità coordinata per i target Lustre su tutti i nodi server OSS/MDS. Mantengono l'appartenenza al cluster e il quorum, monitorano le risorse, orchestrano il failover ordinato e isolano i nodi non responsivi prima di attivare i target condivisi altrove. Questo coordinamento impedisce l'accesso split-brain ai volumi E-Series e protegge l'integrità del file system durante i guasti.
NVIDIA DOCA-OFED	Fornisce driver HCA, librerie RDMA e utilità di gestione per InfiniBand e le fabric RoCE. Lo stesso stack software supporta l'accesso backend NVMe-oF a bassa latenza agli array E-Series e la comunicazione frontend LNet ad alto throughput con i client Lustre.

NetApp fornisce pacchetti RPM precompilati di Lustre server per Rocky Linux 9.8 e Red Hat Enterprise Linux 9.8.

Automazione Ansible

Lustre con NetApp E-Series Storage viene fornito e distribuito tramite automazione Ansible da un nodo di controllo con accesso di gestione agli array E-Series e ai nodi server OSS/MDS. Un inventario Ansible modella la soluzione desiderata, inclusi array, server, assegnazioni di storage, interfacce di rete e risorse del cluster HA.

La NetApp E-Series Lustre Ansible collection orchestra la distribuzione end-to-end utilizzando le SANtricity e Host collections. Insieme, queste collections effettuano il provisioning e la mappatura dello storage E-Series, preparano il sistema operativo del server, configurano la connettività NVMe-oF e LNet, distribuiscono i target Lustre e creano le risorse Pacemaker. Questa automazione rende la configurazione ripetibile e semplifica l'aggiunta di building block a un file system esistente.

La tabella seguente riassume i principali pacchetti software utilizzati sul nodo di controllo Ansible. Consulta la ["Collezione Ansible NetApp E-Series Lustre"](#) per le dipendenze dei pacchetti correnti.

Pacchetto	Dipendenza o scopo
Pitone	Fornisce il runtime per Ansible e i moduli Python necessari.
ansible-core	Richiede Python ed esegue le raccolte Ansible NetApp E-Series.
Pacchetti Python aggiuntivi	cryptography, ipaddr, netaddr, passlib, resolvelib, jinja2, e pyyaml
Collezione Ansible NetApp E-Series Lustre	Configura Lustre, LNet, connettività host NVMe-oF e le risorse Pacemaker.
Collezione Ansible NetApp E-Series SANtricity	Configura array E-Series, pool di storage, volumi e mappature host tramite la SANtricity Web Services API.
Collezione Ansible Host NetApp E-Series	Prepara i nodi server OSS/MDS per la connettività LNet e NVMe-oF agli array E-Series.

Software client Lustre

Il software client Lustre consente a un sistema di accedere al file system parallelo tramite il fabric LNet frontend. Ogni client necessita di moduli kernel Lustre e utility client Lustre compatibili sia con il kernel in esecuzione che con la versione di Lustre distribuita sui server.

Usa "[Matrice di supporto Lustre](#)" per trovare un sistema operativo client e un kernel compatibili con la tua versione di Lustre. NetApp non fornisce pacchetti client Lustre precompilati. Crea i pacchetti client per il sistema operativo client e il kernel dal codice sorgente Lustre fornito da NetApp in "[repository netapp-lustre](#)". Dopo che i pacchetti client Lustre sono stati installati, il ruolo `lustre_client` può configurare le interfacce di rete, LNet e creare un'unità di montaggio `systemd` persistente per il file system Lustre.

Per le istruzioni relative alla distribuzione del client, vedi "[Distribuisce la soluzione](#)".

Lustre con NetApp E-Series Storage - Guida al dimensionamento

Dimensiona Lustre con i building block di storage NetApp E-Series in base alle esigenze di capacità e metadati, decidi quando aggiungere capacità e analizza i fattori che influiscono sulle prestazioni.

Dimensionamento della capacità

Un building block con due array EF80 fornisce il seguente layout target Lustre. Per il posizionamento target preferito e i percorsi NVMe-oF, vedi "[Distribuzione del volume del building block primario](#)" in "[Componenti hardware](#)".

Componente	Conteggio	Dimensione tipica del volume (per target)
MGS	1	5-10 GiB
MDT	8	Le dimensioni variano in base alla capacità dell'unità e al layout RAID
OST	32	Le dimensioni variano in base alla capacità dell'unità e al layout RAID

Ogni array EF80 utilizza ventiquattro unità NVMe. Le capacità supportate sono 3,84 TB, 7,68 TB e 15,3 TB con gruppi di volumi tradizionali (RAID 1 per MGS/MDT e RAID 6 per OST) o DDP, e unità Capacity Flash (QLC) da 30,7 TB o 61,4 TB solo con DDP. Le unità da 1,92 TB non sono attualmente consigliate per questa soluzione. Per la disposizione degli slot delle unità e la selezione del pool, vedi ["Componenti hardware"](#).

La tabella seguente elenca la capacità utilizzabile approssimativa per un building block (due array, 48 drive). La capacità utilizzabile dell'array non corrisponde alla capacità utilizzabile finale del file system Lustre. L'allocazione degli inode MDT, i journal, l'overprovisioning e le percentuali di volume dell'inventario riducono la capacità disponibile per le applicazioni.

Capacità del disco	Layout (per array)	Utilizzabile approssimativo (building block)
3,84 TB	RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10)	138,08 TB
3,84 TB	DDP (24 unità, 2 riservate)	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	Solo DDP	1.070,35 TB
61,4 TB	Solo DDP	2.162,70 TB

Dimensiona metadati e dati separatamente, poi imposta le dimensioni dei volumi nell'inventario Ansible. I modelli di distribuzione formattano i target con `ldiskfs`; MDT utilizza il rapporto inode predefinito di Lustre e i modelli OST impostano `-i 4096`.

- **Metadati (MDT):** Dimensiona gli MDT in base al numero di file previsto. Pianifica circa il doppio del numero di file previsto per la crescita. Un MDT sottodimensionato può impedire la creazione di nuovi file anche quando la capacità OST è ancora disponibile.
- **Dati (OST):** Dimensiona gli OST in base alla capacità utilizzabile e alle esigenze di throughput.
- **MGS:** Usa un piccolo management target (5-10 GiB) solo per la configurazione del file system.

Regola le dimensioni dei volumi MDT e OST nell'inventario per la capacità dell'unità selezionata. Modifica `format_options.mkfsoptions` in `eseries_lustre_filesystem_mdt` o `eseries_lustre_filesystem_ost` solo quando un sito ha bisogno di un rapporto inode diverso. Per informazioni di base sui rapporti inode, vedi ["Lustre Wiki: Messa a punto di Lustre"](#). Per sapere quando aggiungere building block, vedi [Scalabilità](#).

Scalabilità

Aggiungi building block quando la capacità o il carico dei metadati lo richiedono:

- **Solo OST:** Il numero di file e il carico di metadati rientrano nella capacità MDT esistente e hai bisogno di maggiore capacità dati o throughput aggregato. Aggiunge 32 OST e due nodi server OSS/MDS.
- **MDT+OST:** Il numero di file o il tasso di metadati si sta avvicinando alla capacità dell'MDT, oppure hai bisogno sia di un maggiore throughput dei metadati che di una maggiore capacità dei dati. Aggiunge 8 MDT e 32 OST.

Usa `mdt.*.md_stats` sugli MDT esistenti per verificare se i metadati sono saturi. Limita ogni cluster Pacemaker/Corosync a cinque building block (dieci nodi server OSS/MDS). Per implementazioni più grandi, suddividi i building block in modo uniforme tra più cluster HA. Vedi ["Architettura della soluzione"](#) per le regole di scaling.

L'esempio seguente mostra un file system con un building block di base più un building block solo OST in un cluster HA.

Building block	Tipo	Server	Array	MGS	MDT	OST
BB1	Base	2	2	1	8	32
BB2	Solo OST	2	2	0	0	32
Totale		4	4	1	8	64

BB2 registra i suoi target OST rispetto all'MGS in BB1 utilizzando `mgsnode=`. Gli indici OST per BB2 vengono assegnati globalmente (ad esempio, da OST0032 a OST0063) in modo che nessun indice sia in conflitto con BB1.

Prestazione

La validazione formale utilizza IOR, `mdtest` e `fio` dai client Lustre per caratterizzare il throughput, gli IOPS e il comportamento dei metadati, e per convalidare il failover HA. Questi test non pubblicano numeri di performance garantite. I benchmark sintetici misurano il comportamento nel caso migliore e potrebbero non riflettere le performance reali dell'applicazione.

Le performance scalano approssimativamente con il numero di building block. I risultati effettivi dipendono dal tipo di drive e dalla configurazione del pool, dal numero di percorsi NVMe-oF, dal fabric LNet e dal numero di client, e dalla combinazione di dati rispetto a metadati I/O.

Contatta il tuo team account NetApp per ricevere consigli specifici sul dimensionamento e sulle prestazioni in base al carico di lavoro.

Per la configurazione delle unità e il conteggio dei volumi, vedi ["Componenti hardware"](#). Per i passaggi di implementazione, vedi ["Distribuisci la soluzione"](#). Per le linee guida sull'inventario a livello di campo, vedi ["Personalizza l'inventario di Lustre Ansible"](#).

Distribuisci la soluzione

Implementa Lustre con NetApp E-Series Storage

Implementa Lustre con NetApp E-Series Storage completando le attività di hardware, preparazione dell'ambiente, inventario, deployment con Ansible e verifica, in quest'ordine.

Prerequisiti

Prima di avviare la distribuzione, assicurati di quanto segue:

- Hai seguito le indicazioni relative alle dimensioni e ai building block in ["Guida alle taglie"](#) e hai selezionato l'hardware che corrisponde a ["Componenti hardware"](#).

- Hai confermato che la combinazione di server, HCA, E-Series, sistema operativo, Lustre, SANtricity OS e DOCA-OFED è elencata per **E-Series Lustre** nel "[Tool NetApp Interoperability Matrix](#)".
- Tutti i nodi server OSS/MDS e gli array E-Series per i building block previsti sono in loco e pronti per essere installati nei rack.
- Hai le credenziali e l'accesso alla rete per i BMC dei server, le porte di gestione dell'array e l'host che fungerà da nodo di controllo Ansible.



NetApp supporta la distribuzione del cluster HA Lustre e dei client Lustre con "[Collezione Ansible NetApp E-Series Lustre](#)". Non configurare manualmente i target Lustre, LNet, la connettività host NVMe-oF o le risorse Pacemaker.

Flusso di lavoro di distribuzione

Completa le seguenti attività in ordine:

1. "[Hardware Lustre rack e cavi](#)".

Installa i server Lustre e gli array EF80 nei rack, poi collega i backend NVMe-oF e i frontend LNet.

2. "[Prepara l'ambiente di deployment di Lustre](#)".

Configura gli array e i server, quindi installa Ansible e le sue dipendenze sul nodo di controllo.

3. "[Personalizza l'inventario di Lustre Ansible](#)".

Seleziona il modello per il fabric del sito e inserisci i server, gli array, le reti, i building block e le credenziali.

4. "[Distribuisci il cluster HA Lustre e i client](#)".

Prepara il sistema operativo del server, installa NVIDIA DOCA-OFED, esegui il playbook di distribuzione principale e configura i client Lustre.

5. "[Verifica ed espandi l'implementazione di Lustre](#)".

Conferma l'integrità del cluster, i mount e la capacità prima dell'uso in produzione. Estendi l'inventario esistente quando il file system ha bisogno di un altro building block.

Informazioni correlate

- "[Collezione Ansible NetApp E-Series Lustre](#)"
- "[Repository sorgente NetApp Lustre](#)"
- "[Guida utente all'amministrazione HA](#)"
- "[Guida all'ottimizzazione delle prestazioni](#)"
- "[Modelli di distribuzione client Lustre](#)"

Hardware Lustre per rack e cavi

Installa nei rack i server Lustre e gli array NetApp EF80, poi collega i fabric NVMe-oF di backend e i fabric LNet di frontend per ciascun building block.

Prima di iniziare

Esamina i conteggi HCA, i requisiti degli slot PCIe, i requisiti del percorso e le linee guida MTU in ["Componenti hardware"](#).

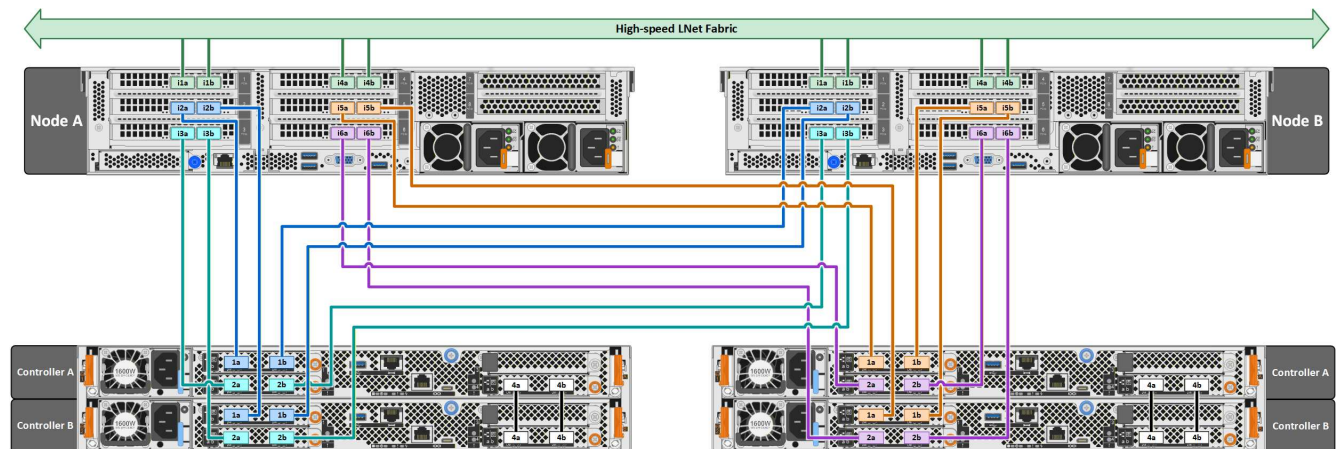
Passaggi

1. Racca e collega tutti i server Lustre e gli storage E-Series secondo la topologia del building block.



2. Per ciascun array EF80, collega la porta 4a del controller A alla porta 4a del controller B e la porta 4b del controller A alla porta 4b del controller B. Queste connessioni dedicate forniscono il mirroring tra controller e non vengono utilizzate per il traffico host. Vedi ["Collega i cavi di mirroring tra i controller EF50 ed EF80"](#).
3. Collega la rete NVMe-oF di backend tra server e array secondo la topologia EF80 six-HCA.

Cablaggio backend EF80 a sei HCA (RoCE o InfiniBand)



4. Collega le quattro interfacce frontend LNet (gli HCA i1 e i4) su ciascun server Lustre e sui client Lustre al fabric di switch dedicato InfiniBand o RoCE.
5. Quando usi RoCE, configura il fabric per il funzionamento lossless con il priority flow control (PFC). I playbook Ansible configurano RoCE lossless su tutte le interfacce nel building block.

Dopo aver finito

["Prepara i server, gli array e il nodo di controllo Ansible"](#).

Prepara l'ambiente di deployment di Lustre

Configura l'accesso di gestione e le impostazioni di base su ciascun array EF80 e server Lustre, quindi prepara un nodo di controllo Ansible per distribuire la soluzione.

Prepara gli array E-Series

Passaggi

1. Configura l'interfaccia di gestione su ciascun controller e verifica che SANtricity System Manager sia raggiungibile.
2. Imposta il nome dell'array e le credenziali amministrative che l'inventario Ansible userà.
3. Conferma che l'array sia in uno stato ottimale e che tutte le unità siano presenti e segnalino correttamente.

Per istruzioni sulla configurazione della connessione di gestione, vedi ["Documentazione E-Series"](#).

Prepara i server Lustre

Passaggi

1. Configura il controller di gestione della scheda madre (BMC) su ogni server per l'accesso fuori banda e imposta le impostazioni di sistema UEFI per le massime prestazioni.
2. Installa un sistema operativo supportato (RHEL o Rocky Linux), quindi configura l'host name e la porta di rete di gestione.
3. Prepara tutti i pacchetti firmware e software per soddisfare i requisiti della soluzione **E-Series Lustre**, come descritto in ["Componenti software"](#).

Prepara il nodo di controllo Ansible

Designa un sistema Linux virtuale o fisico con accesso di rete di gestione a tutti i server Lustre e agli array E-Series come nodo di controllo Ansible.

I seguenti passaggi utilizzano Ubuntu 24.04. Per le istruzioni relative a un'altra distribuzione Linux, vedi la ["documentazione di installazione di Ansible"](#).

Passaggi

1. Installa Python, pip e il pacchetto per l'ambiente virtuale Python:

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

2. Crea e attiva un ambiente virtuale Python:

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

3. Installa Ansible e i pacchetti Python necessari nell'ambiente virtuale attivato:

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

```
passlib pyyaml resolvelib
```

4. Installa la NetApp E-Series Lustre Ansible collection. L'installazione della collection installa anche le sue collection dipendenti, incluse le collection SANtricity e Host:

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

5. Verifica le versioni di Ansible, Python e della collection installate:

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

6. Configura l'accesso SSH senza password dal nodo di controllo a ciascun server Lustre. Sostituisci <ip_or_hostname> con l'indirizzo IP di gestione o l'host name di un server:

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

Ripeti `ssh-copy-id` per ogni server Lustre.



Non configurare l'accesso SSH senza password agli array E-Series. Ansible gestisce gli array tramite la SANtricity Web Services API utilizzando le credenziali definite nell'inventario.

Dopo aver finito

["Personalizza l'inventario di Ansible"](#).

Personalizza l'inventario Lustre Ansible

Crea una copia funzionante di un modello di distribuzione EF80 e personalizza il suo inventario per i tuoi server, array, reti e building block Lustre.

Mantieni l'inventario selezionato nella sua directory fabric e piattaforma. Mantieni la directory condivisa `playbooks` e il file `passwords.yml` nella radice di `building_blocks`.

Seleziona un modello di distribuzione

Parti dal modello che corrisponde al fabric di rete del sito:

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

Configura le impostazioni specifiche del sito

Imposta i valori specifici del sito nell'inventario, insieme alle impostazioni del repository locale e delle regole udev. I riferimenti di riga nella tabella puntano ai file modello RoCE EF80 nel `ansible-lustre release/1.0.0` tag. Il modello InfiniBand utilizza gli stessi nomi di variabili e la stessa struttura, ad eccezione della variabile di interfaccia LNet, che è indicata nella tabella.

Campo	variabile Ansible	File modello (release/1.0.0)	Ambito	Note
Nome file system	<code>eseries_lustre_filesystem_mgt.format_options.fsname,</code> <code>eseries_lustre_filesystem_mdt.format_options.fsname,</code> <code>eseries_lustre_filesystem_ost.format_options.fsname</code>	"ha_cluster.yml#L76" , "#L90" , "#L110"	a livello di cluster	Annidato sotto <code>format_options</code> per le mappe MGT, MDT e OST. Massimo 8 caratteri. Usa lo stesso valore in tutte e tre le posizioni. Una chiave <code>fsname</code> di primo livello non ha effetto.
NID MGS per la registrazione dei target	<code>eseries_lustre_filesystem_mdt.format_options.mgsnode,</code> <code>eseries_lustre_filesystem_ost.format_options.mgsnode</code>	"ha_cluster.yml#L98" , "#L118"	a livello di cluster	Annidato sotto MDT e OST <code>format_options</code> . Includi gli NID LNet frontend di entrambi i nodi nella coppia HA. Elenca prima gli NID del nodo configurato come proprietario MGS preferito, seguiti dagli NID del nodo partner.
Nome cluster Pacemaker	<code>eseries_lustre_pacemaker_cib_properties_overrides.cluster_properties.cluster-name</code>	"ha_cluster.yml#L228"	a livello di cluster	Annidato sotto le sovrascritture delle proprietà CIB di Pacemaker. Nome univoco per il cluster HA.
Indirizzi di fencing BMC	<code>eseries_lustre_pacemaker_fencing_agents.fence_redfish.ip</code>	"ha_cluster.yml#L211" , "#L214" , "#L217" , "#L220"	Ripeti per server Lustre	Indirizzo IP Redfish BMC per ciascun nodo server.
Destinatari email di avviso	<code>eseries_lustre_alert_email_list</code>	"ha_cluster.yml#L233"	a livello di cluster	Destinatari delle notifiche relative alle risorse HA e ai guasti.
Dominio mail	<code>eseries_lustre_mail_service_overrides.conf.mydomain</code>	"ha_cluster.yml#L240"	a livello di cluster	Dominio Postfix usato per inviare l'email di avviso.

Campo	variabile Ansible	File modello (release/1.0.0)	Ambito	Note
Sorgenti orarie NTP	<code>eseries_lustre_time_sync_overrides.server_pools</code>	" ha_cluster.yml#L250 "	a livello di cluster	Sorgenti orarie per i nodi del cluster. Specifica un pool NTP o uno o più server NTP individuali. Includi la direttiva richiesta <code>pool</code> o <code>server</code> e opzioni come <code>iburst</code> .
Mappatura udev da PCI a nome dell'interfaccia	<code>eseries_ip_udev_rules</code>	" ha_cluster.yml#L172 "	Ripeti per layout slot HCA	Mappa ogni slot PCI a un nome di interfaccia persistente (valori a L182). Raccogli con <code>lspci -nnvmm</code> .
repository locale (facoltativo)	<code>eseries_common_yum_repos</code>	" ha_cluster.yml#L256 ", " #L261 "	a livello di cluster	Per le installazioni isolate dalla rete, decommenta il blocco e imposta il <code>repository url</code> (L264).
IP di gestione del server	<code>ansible_host</code>	" host_vars/lustre_01.yml#L10 "	Ripeti per server Lustre	Un <code>host_vars/lustre_NN.yml</code> file per server.
Interfacce backend NVMe-oF	<code>eseries_nvme_roce_interfaces</code> (RoCE) / <code>eseries_nvme_ib_interfaces</code> (InfiniBand)	" RoCE host_vars/lustre_01.yml#L21 ", " InfiniBand host_vars/lustre_01.yml#L17 "	Ripeti per server Lustre	Indirizzi dell'interfaccia backend utilizzati per il traffico di storage NVMe-oF verso gli array NetApp E-Series.
Interfacce del cluster LNet	<code>eseries_roce_interfaces</code> (RoCE) / <code>eseries_ipoib_interfaces</code> (InfiniBand)	" RoCE host_vars/lustre_01.yml#L47 ", " InfiniBand host_vars/lustre_01.yml#L38 "	Ripeti per server Lustre	Indirizzi dell'interfaccia LNet del frontend (valori RoCE a L56; InfiniBand valori a L47).
Indirizzi IP dei nodi Corosync	<code>eseries_lustre_corosync_node_ips</code>	" host_vars/lustre_01.yml#L58 "	Ripeti per server Lustre	Elenca tutti gli indirizzi IP dei nodi nel cluster HA (valori a L62).
NID del servizio target Lustre	<code>lustre_target_service_nodes</code>	" host_vars/lustre_01.yml#L64 "	Ripeti per server Lustre	Includi gli NID LNet frontend (<code>@o2ib</code> da entrambi i nodi della coppia HA così i servizi di destinazione restano raggiungibili dopo il failover (valori a L67).
IP di gestione array	<code>ansible_host</code>	" host_vars/netapp_01.yml#L13 "	Ripeti per ogni array E-Series	Indirizzo IP di gestione di un controller; l'URL dell'API deriva da esso.

Campo	variabile Ansible	File modello (release/1.0.0)	Ambito	Note
Appartenenza al gruppo dei building block	mgs / mds_NN / oss_NN	"lustre_inventory.yml"	Per building block	Ogni nome host dell'inventario deve corrispondere al nome base del file <code>host_vars</code> corrispondente. Ad esempio, l'host <code>lustre_01</code> usa <code>host_vars/lustre_01.yml</code> , e <code>netapp_01</code> usa <code>host_vars/netapp_01.yml</code> . Includi il gruppo MGS solo nel primo building block. Includi i gruppi MDS nel building block di base e in ogni building block di espansione MDT+OST; ometti i gruppi MDS solo dai building block di espansione solo OST. Registra le destinazioni di espansione rispetto all'MGS esistente con <code>mgsnode</code> .

Ripeti le impostazioni per ogni host e building block. Crea un `host_vars/lustre_NN.yml` file per ogni server Lustre e un `host_vars/netapp_NN.yml` file per ogni array, e ripeti i gruppi di inventario per ogni building block aggiuntivo.

I playbook di distribuzione caricano le credenziali di array, Pacemaker e BMC dal file condiviso `building_blocks/passwords.yml`. Popola questo file e crittografalo con Ansible Vault prima della distribuzione, oppure sovrascrivi i valori in modo sicuro in fase di esecuzione con `--extra-vars`. Non inserire le credenziali in chiaro nel controllo della versione.



Gli indirizzi IP, i nomi host, i nomi delle interfacce e il dimensionamento dei volumi nei modelli di deployment sono esempi. Modifica tutti i file di inventario per l'ambiente prima di eseguire qualsiasi playbook.

Dopo aver finito

"Distribuisci il cluster HA Lustre e i client".

Distribuisci il cluster Lustre HA e i client

Prepara il sistema operativo del server, installa NVIDIA DOCA-OFED, distribuisci il cluster HA Lustre e configura i client Lustre utilizzando i playbook Ansible di NetApp.

Prima di iniziare, completa e conserva l'inventario descritto in "[Personalizza l'inventario di Lustre Ansible](#)".

Distribuisci il cluster HA Lustre

Passaggi

1. Dalla `building\_blocks/playbooks` directory nella tua copia di lavoro, prepara il sistema operativo sui server Lustre. Sostituisci `` con il percorso della tua directory di inventario personalizzata:

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

2. Installa NVIDIA DOCA-OFED su ogni server Lustre. Compila i moduli del kernel con il kernel che hai installato nel passaggio precedente. Segui la procedura per il tuo kernel in ["Installazione e aggiornamento di NVIDIA DOCA-Host"](#). Il seguente esempio mostra un'installazione di DOCA-OFED su RHEL o Rocky Linux.

- a. Installa il pacchetto repository host DOCA e aggiorna la cache dei pacchetti. Sostituisci `<doca_host_package>` con il pacchetto host DOCA per il tuo sistema operativo e architettura:

```
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. Compila i moduli del kernel DOCA per il kernel in esecuzione. Lo script stampa il percorso del pacchetto che crea:

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

- c. Installa il pacchetto repository del modulo kernel creato dallo script, quindi aggiorna la cache dei pacchetti. Sostituisci `<doca_kernel_repo>` con il percorso del passaggio precedente:

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

- d. Installa lo spazio utente DOCA-OFED, il kernel e i pacchetti NVMe. Sostituisci `<kernel_version>` con la versione del kernel in esecuzione:

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

3. Esegui il playbook di distribuzione principale:

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



Regola `--forks` in base alle dimensioni della distribuzione per controllare quanti host Ansible configura in parallelo. Ad esempio, aggiungi `--forks 20` per un cluster più grande.

Il playbook esegue il provisioning di gruppi di volumi E-Series o pool e volumi DDP, configura la connettività host NVMe-oF, formatta e monta i target Lustre, configura LNet, applica l'ottimizzazione delle prestazioni e configura le risorse HA di Pacemaker.



Un'implementazione con un singolo building block forma un cluster a due nodi Pacemaker. Al nodo A viene assegnato un voto aggiuntivo in caso di failover. Per raggiungere il quorum in un cluster a due nodi, valuta la configurazione di un qdevice. Quando un'implementazione contiene più building block, ciascun building block contribuisce con una coppia di server a due nodi al cluster Pacemaker più grande, fino al limite di cluster documentato. Esamina la configurazione di fencing e quorum in ["Guida utente all'amministrazione HA"](#) così il cluster può ripristinare in sicurezza i target durante un errore di un nodo.

Distribuisci i client Lustre

Distribuisci i client Lustre su qualsiasi sistema che richieda l'accesso al file system personalizzando il modello di distribuzione del client ed eseguendo il playbook del client.

Passaggi

1. Seleziona un sistema operativo client e un kernel compatibili da ["Matrice di supporto Lustre"](#). Compila e installa i pacchetti client Lustre per quel kernel da ["netapp-lustre"](#) se non sono già installati.
2. Crea una copia di lavoro della directory dei `clients` template, incluso il file `passwords.yml` condiviso, e seleziona il template che corrisponde al tuo fabric:
 - **InfiniBand:** ["IB_lustre_cluster_clients"](#)
 - **RoCE:** ["ROCE_lustre_cluster_clients"](#)
3. Personalizza `lustre_client_inventory.yml`, `host_vars` e `group_vars` per i tuoi clienti. I riferimenti di riga nella tabella puntano ai file modello client RoCE nel tag `ansible-lustre release/1.0.0`; il modello client InfiniBand utilizza le stesse variabili, salvo diversa indicazione.

Campo	variabile Ansible	File modello (release/1.0.0)	Ambito	Note
Host client	<code>lustre_clients</code>	"lustre_client_inventory.yml#L4"	Per client	Elenca ogni host name del client.
Utente SSH del client e password di accesso	<code>ssh_client_user</code> / <code>ssh_client_become_pass</code>	"passwords.yml#L2"	a livello di cluster	Imposta la password become solo se l'utente SSH non è root.
Interfacce LNet	<code>eseries_lustre_lnet</code>	"group_vars/all.yml#L7"	a livello di cluster	Nomi delle interfacce LNet lato client (valori a L13).

Campo	variabile Ansible	File modello (release/1.0.0)	Ambito	Note
Monta NID MGS	lustre_client_mounts.mgsnode	"group_vars/all.yml#L17"	a livello di cluster	NID MGS utilizzati per montare il file system (valori a L20).
Nome file system	lustre_client_mounts.fsname	"group_vars/all.yml#L21"	a livello di cluster	Deve corrispondere al cluster fsname.
Punto di mount	lustre_client_mounts.mount_point	"group_vars/all.yml#L22"	a livello di cluster	Directory di mount del client.
IP di gestione client	ansible_host	"host_vars/client_01.yml#L4"	Ripeti per client	Un `host_vars/client_NN.yml` file per client.
Interfacce fabric di storage	eseries_roce_interfaces	"host_vars/client_01.yml#L10"	Ripeti per client	Indirizzi dell'interfaccia frontend (valori a L15). Il template InfiniBand utilizza eseries_ipoib_interfaces qui.

Ripeti le impostazioni per ogni client. Crea un `host_vars/client_NN.yml` file per ogni client e aggiungi l'host corrispondente a `lustre_client_inventory.yml`. Popola il file condiviso `clients/passwords.yml` e crittografalo con Ansible Vault prima di eseguire il playbook del client.

4. Esegui il playbook del client dalla directory del template del client:

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

Il playbook del client configura le interfacce di rete del client e LNet, e può creare un'unità di montaggio systemd persistente per il file system Lustre.

Dopo aver finito

["Verifica l'implementazione"](#).

Verifica ed espandi la distribuzione di Lustre

Conferma l'integrità del cluster, i mount di Lustre e la capacità del file system prima dell'uso in produzione, poi usa lo stesso inventario per aggiungere building block quando necessario.

Verifica l'implementazione

Passaggi

1. Da un server Lustre, verifica che tutte le risorse Pacemaker siano avviate sui nodi previsti:

```
pcs status
```

2. Da un client Lustre, verifica che il file system Lustre sia montato:

```
mount -t lustre
```

3. Da un client Lustre, controlla la capacità del file system e che i target MDT e OST siano visibili:

```
lfs df -h
```

Per la convalida delle prestazioni con IOR, mdtest e fio, vedi ["Guida alle taglie"](#).

Per la migrazione controllata della destinazione o per i test di failover HA, segui le procedure nel ["Guida utente all'amministrazione HA"](#).

Espandi il file system

Per aggiungere capacità o prestazioni dei metadati, estendi il tuo inventario esistente con un ulteriore building block. Non creare un inventario separato per il nuovo building block. Il playbook assegna automaticamente indici MDT e OST univoci e registra i nuovi target sull'MGS esistente.

Passaggi

1. Aggiungi i nuovi building block host, array e gruppi di risorse (MDT+OST o solo OST) allo stesso inventario e `host_vars` e `group_vars` file che hai usato per la distribuzione originale.
2. Esegui di nuovo il playbook di distribuzione principale sull'inventario aggiornato.

Vedi ["Architettura della soluzione"](#) per le regole di scaling e ["Guida alle taglie"](#) per indicazioni su quando aggiungere ciascun tipo di building block.

Lustre con NetApp E-Series Storage - Risorse correlate

Usa questi link per la documentazione, gli strumenti e il software correlati mentre pianifichi o espandi una distribuzione di Lustre con NetApp E-Series Storage. Per informazioni su dimensionamento e passaggi di implementazione, vedi ["Guida alle taglie"](#) e ["Distribuisci la soluzione"](#).

Risorse NetApp

- ["Tool NetApp Interoperability Matrix"](#). Cerca **E-Series Lustre**.
- ["NetApp EF-Series array all-flash datasheet"](#)
- ["NetApp documentazione di SANtricity System Manager"](#)
- ["collezione NetApp ansible-lustre"](#). Per le linee guida relative ai campi di inventario, vedi ["Personalizza l'inventario di Lustre Ansible"](#).
- ["Repository sorgente NetApp Lustre"](#). NetApp fornisce pacchetti RPM precompilati per il server Lustre per Rocky Linux 9.8 e Red Hat Enterprise Linux 9.8. Compila i pacchetti client Lustre per il sistema operativo e

il kernel client a partire dai sorgenti NetApp.

Community Lustre

- ["file system Lustre"](#)
- ["Matrice di supporto Lustre"](#) per sistemi operativi client e kernel
- ["Lustre Wiki: Messa a punto di Lustre"](#)

Soluzioni di infrastruttura AI correlate NetApp

- ["BeeGFS su NetApp con storage E-Series"](#)
- ["Implementazione di IBM Spectrum Scale con NetApp E-Series Storage"](#)
- ["NVIDIA DGX SuperPOD con NetApp EF-Series"](#)

Informazioni sul copyright

Copyright © 2026 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.