# Lustre con almacenamiento E-Series de NetApp

NetApp artificial intelligence solutions

NetApp
August 31, 2026

# Tabla de contenidos

# Lustre con almacenamiento E-Series de NetApp

## Lustre con el almacenamiento NetApp E-Series: introducción

Lustre con el almacenamiento E-Series de NetApp es una solución validada de sistema de archivos paralelo para cargas de trabajo de IA y HPC, construida a partir de bloques repetibles de pares de servidores de alta disponibilidad y arrays all-flash E-Series.

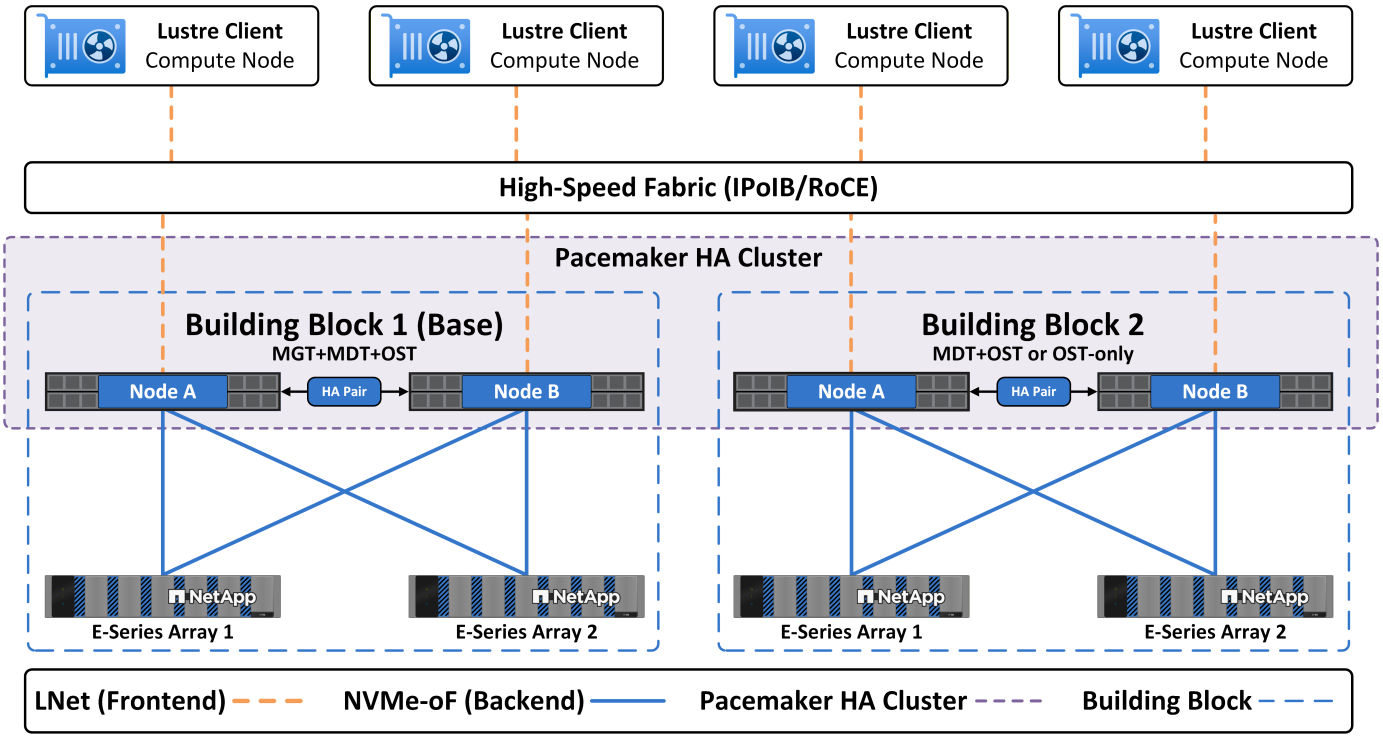
### Descripción general de la solución

Lustre con el almacenamiento E-Series de NetApp combina el sistema de archivos paralelo de código abierto Lustre con el almacenamiento en bloques E-Series de NetApp para ofrecer una base escalable y de alto rendimiento para cargas de trabajo con un uso intensivo de datos. Cada módulo básico combina dos nodos de servidor OSS/MDS configurados para alta disponibilidad (HA) con dos E-Series all-flash arrays. La conectividad NVMe-oF del backend transmite las operaciones de E/S de bloques a las arrays y una estructura LNet independiente conecta a los clientes y a los nodos de servidor OSS/MDS para permitir el acceso paralelo al sistema de archivos.

Un bloque de construcción básico aloja el servidor de gestión (MGS) y los destinos de metadatos iniciales (MDT) y los destinos de almacenamiento de objetos (OST). Los bloques de construcción adicionales se registran en el MGS básico para ampliar la capacidad de metadatos, la capacidad de datos y el rendimiento agregado a medida que aumentan las demandas de la carga de trabajo.

La siguiente figura muestra un ejemplo de implementación con varios bloques de construcción y clientes Lustre conectados a través de LNet.

**Descripción general de la solución de almacenamiento NetApp E-Series con Lustre**



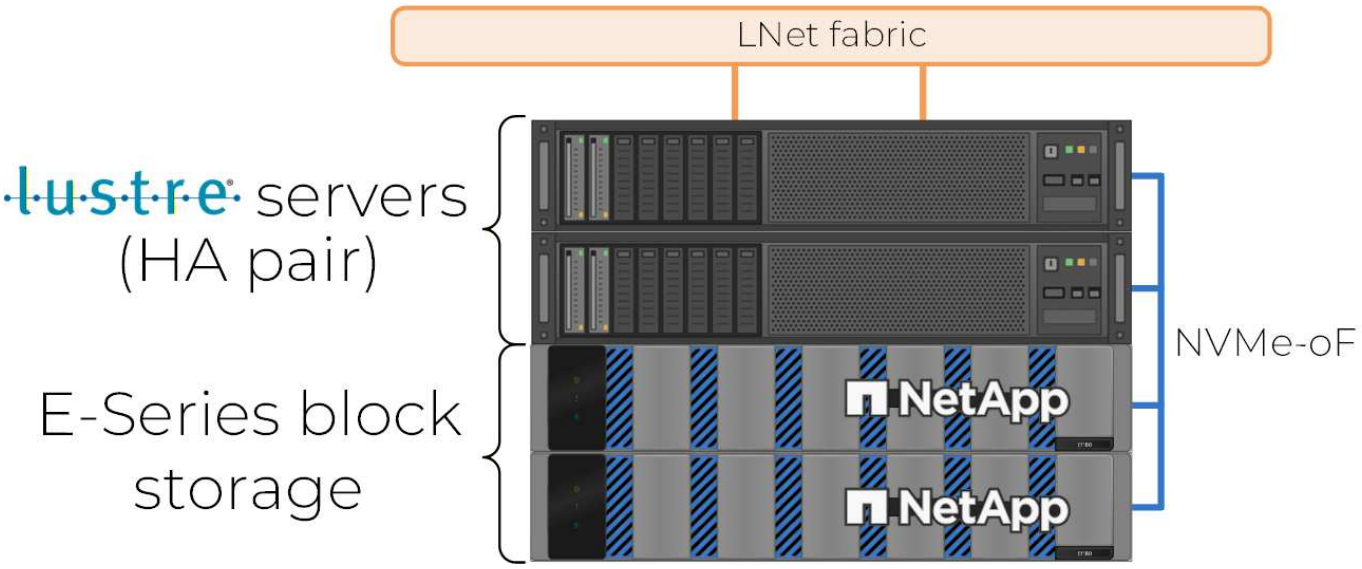
# Lustre con almacenamiento E-Series de NetApp - arquitectura de la solución

Descubre cómo Lustre con el almacenamiento E-Series de NetApp utiliza módulos, alta disponibilidad, topología de red y clientes para que puedas planificar la capacidad, el rendimiento y la conmutación por error.

## Diseño modular

Un building block es la unidad fundamental y escalable de la solución Lustre E-Series de NetApp. Cada building block consta de dos nodos de servidor configurados como un par de alta disponibilidad (HA) que proporcionan funciones de servidor de almacenamiento de objetos Lustre (OSS) y de servidor de metadatos (MDS), dos arrays all-flash E-Series de NetApp, conectividad dedicada de backend NVMe over Fabrics (NVMe-oF) entre servidores y arrays, y conectividad dedicada de red Lustre (LNet) de frontend para clientes y comunicación entre nodos. Un clúster HA de Pacemaker y Corosync puede contener el par de servidores de uno o más building blocks. La colección NetApp `ansible-lustre` utiliza la automatización de Ansible para implementar y configurar los servicios de Lustre.

**Módulo básico de Lustre**




Los bloques de construcción se adaptan a los requisitos de crecimiento del sistema de archivos. Cada sistema de archivos requiere un bloque de construcción base. El bloque de construcción base aloja el servidor de gestión (MGS) en un destino de gestión (MGT) y proporciona capacidad inicial para el destino de metadatos (MDT) y el destino de almacenamiento de objetos (OST). Los bloques de construcción adicionales amplían el sistema de archivos y registran sus destinos MDT y OST en el MGS del bloque de construcción base. El enfoque modular permite que la capacidad y el rendimiento se escalen de forma independiente en función de las necesidades de la carga de trabajo.

NetApp recomienda los siguientes perfiles de configuración de bloques de construcción para la mayoría de las implementaciones. Los recuentos de MGS, MDT y OST indicados son valores predeterminados recomendados, optimizados para maximizar el rendimiento y la capacidad de las matrices de almacenamiento E-Series. Ajusta los recuentos de destinos, los tamaños de los volúmenes y la asignación de unidades E-Series en el inventario de Ansible para adaptarlos a las necesidades de la carga de trabajo. Por ejemplo, las implementaciones que requieren una mayor capacidad de almacenamiento de metadatos pueden aprovisionar más MDT y menos OST dentro del mismo bloque de hardware.

La siguiente tabla resume los tipos de bloques de construcción y los recuentos objetivo recomendados.

| Tipo | MGS | MDTs | OST | Uso |
|---|---|---|---|---|
| Base | 1 | 8 | 32 | Primer componente básico de un sistema de archivos. Alberga el MGS y la capacidad inicial de MDT y OST. |
| MDT+OST | 0 | 8 | 32 | Añade metadatos y capacidad de datos. Regístralo contra el bloque de construcción base MGS. |
| Solo OST | 0 | 0 | 32 | Solo añade capacidad de datos. Regístrate contra el bloque de construcción base MGS. |

- **Tipo de unidad y estructura del grupo:** E-Series es compatible con grupos de volúmenes RAID tradicionales y Dynamic Disk Pools (DDP). Para las unidades NVMe TLC, usa grupos de volúmenes RAID 6 para el almacenamiento OST y grupos de volúmenes RAID 1 para el almacenamiento MGS y MDT. Para las unidades NVMe QLC, usa DDP para el grupo de unidades compartido.
- **Distribución de objetivos:** Los objetivos de Lustre en cada bloque de construcción se distribuyen de forma equilibrada entre ambos nodos de servidor OSS/MDS y ambas matrices E-Series. La distribución reparte las operaciones de E/S entre las CPU de los servidores y los controladores de las matrices para mejorar el rendimiento de las rutas NVMe-oF y mantener la redundancia. Consulta "distribución de destino y conectividad NVMe-oF" en "Componentes de hardware".

Los sistemas operativos compatibles, las versiones de Lustre, el firmware de los arrays y los componentes básicos relacionados se enumeran en "Herramienta de matriz de interoperabilidad (IMT) de NetApp" bajo **E-Series Lustre**. Para obtener información detallada sobre el recuento de volúmenes, la disposición de las unidades y las directrices de ajuste de tamaño, consulta "Componentes de hardware". Para los componentes de software, consulta "Componentes de software".

## Recomendaciones de escalado

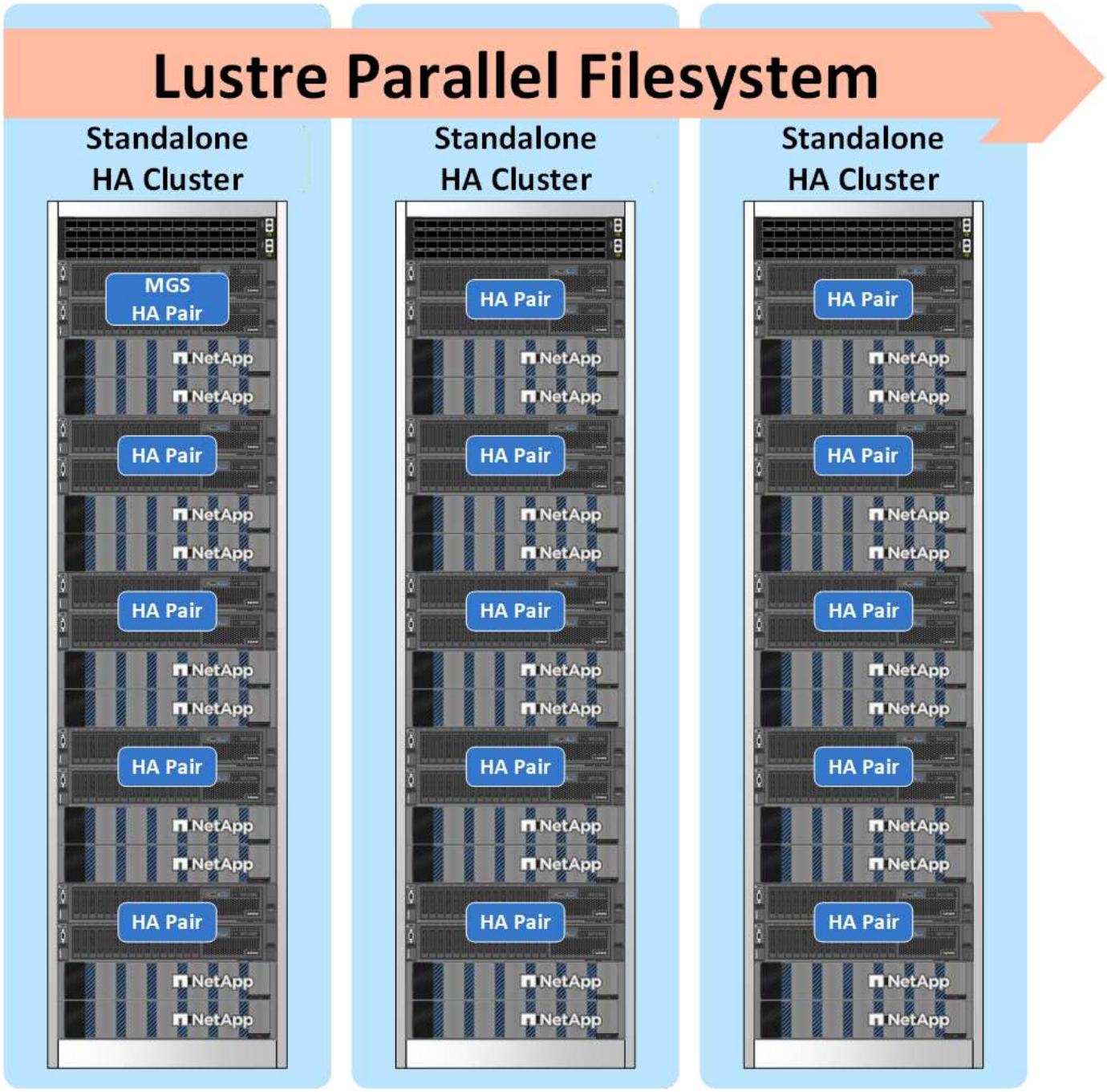
El sistema de archivos Lustre se amplía añadiendo módulos cuando la capacidad de metadatos, la capacidad de datos o ambas se convierten en un factor limitante. Amplía los MDT en función del número de archivos y las IOPS de metadatos; amplía los OST en función de la capacidad y los requisitos de rendimiento agregado.

1. **Un bloque básico por sistema de archivos:** Implementa exactamente un bloque básico; este alberga el MGS y los destinos iniciales de MDT y OST.
2. **Perfil de módulo adicional:** Añade un módulo «solo OST» cuando la capacidad MDT existente sea suficiente y sea necesario aumentar la capacidad de datos o el rendimiento. Añade un módulo «MDT+OST» cuando el rendimiento de los metadatos o la capacidad del espacio de nombres se conviertan en el cuello de botella.

3. **Límite de clúster HA:** Limita cada clúster HA de Pacemaker y Corosync a cinco bloques de construcción (diez nodos de servidor Lustre). Divide las implementaciones más grandes de forma equitativa en varios clústeres HA para evitar limitaciones de recursos en configuraciones grandes.

La siguiente figura muestra hasta cinco módulos apilados en un rack de 42U (un clúster de Pacemaker HA por rack). Varios racks pueden formar parte de un único sistema de archivos Lustre; solo el módulo base aloja el MGS.

**Escalabilidad de los racks del sistema de archivos paralelo Lustre**




Para ver ejemplos de dimensionamiento, consulta "Guía de tallas".

## Arquitectura de HA

Lustre con el almacenamiento E-Series de NetApp utiliza una arquitectura de alta disponibilidad (HA) con disco compartido integrada con el almacenamiento E-Series de NetApp. Pacemaker gestiona los recursos de alta disponibilidad y Corosync proporciona la pertenencia al clúster y la mensajería. Juntos, gestionan la conmutación por error de los destinos de Lustre entre los dos nodos de servidor OSS/MDS en cada bloque funcional. NVMe-oF multivía conecta ambos nodos de servidor OSS/MDS a los mismos volúmenes E-Series, así que cualquiera de los nodos puede hacerse cargo de los servicios de destino cuando sea necesario.

Cada MGS, MDT y OST se configura como un recurso de alta disponibilidad (HA) con sus dependencias en un grupo de recursos de Pacemaker. Pacemaker garantiza que los recursos se inicien y se detengan en el orden correcto, permanezcan ubicados en el mismo nodo y se ejecuten en un solo nodo a la vez. El acceso simultáneo al mismo destino desde ambos nodos podría dañar el sistema de archivos subyacente.

- **Conmutación por error del destino:** Ambos nodos del servidor OSS/MDS son pares en el clúster, pero cada destino Lustre está activo en un solo nodo a la vez. Un recurso de supervisión de Pacemaker controla el estado de cada destino y sus dependencias y activa una conmutación por error cuando un destino deja de estar disponible en su nodo actual. En caso de fallo, Pacemaker reinicia el grupo de recursos afectado en el nodo asociado en el orden correcto, y los clientes se vuelven a conectar de forma transparente al destino en su nueva ubicación una vez que se reanudan los servicios. Dado que cada destino se activa en un único nodo, el sistema de archivos nunca está expuesto a escrituras simultáneas desde ambos nodos.
- **Acceso al almacenamiento compartido:** Las rutas NVMe-oF multivía conectan cada nodo de servidor OSS/MDS con todos los controladores E-Series del bloque modular, así que se puede acceder a cada volumen de destino desde cualquiera de los nodos. Si un nodo de servidor falla, el nodo asociado ya tiene rutas activas a los mismos volúmenes y puede asumir la gestión de los destinos sin reconfigurar la conectividad de almacenamiento. Si falla un controlador de la matriz o una ruta, multivía redirige las operaciones de E/S a un controlador que siga operativo. Esta doble redundancia permite que los destinos de Lustre hagan conmutación por error entre nodos de servidor OSS/MDS o controladores de la matriz sin perder acceso a los volúmenes de respaldo.
- **Aislamiento STONITH:** Cuando se produce un fallo, Pacemaker a veces no puede comunicarse con el nodo defectuoso para confirmar que sus objetivos están detenidos. Antes de reiniciar esos objetivos en otro lugar, Pacemaker aísla el nodo defectuoso, idealmente cortándole la alimentación, para asegurarse de que está inactivo. El fencing evita una situación de split-brain en la que ambos nodos acceden al mismo objetivo simultáneamente y dañan el sistema de archivos subyacente, preservando la integridad de los datos durante la conmutación por error. NetApp recomienda `fence_redfish` para servidores con controladores de gestión de placa base (BMC) compatibles con Redfish. También se admiten otros agentes de fencing, como `fence_apc`.

Para la administración del clúster y la configuración del fencing, consulta la "Guía de usuario de HA" en la "colección ansible-lustre".

## Arquitectura de red

La solución utiliza rutas de red independientes para el tráfico de backend de almacenamiento y el tráfico de frontend de LNet.

- **Backend (NVMe-oF):** Los nodos de servidor OSS/MDS se conectan a las matrices E-Series a través de NVMe-oF usando NVMe/InfiniBand o NVMe/RoCE. La ruta NVMe-oF transporta E/S de bloques entre los servicios de destino Lustre y los volúmenes E-Series. Para el cableado backend de seis HCA del EF80, consulta "Rack y cableado de hardware Lustre". Para la ubicación recomendada de los destinos entre nodos y matrices, consulta "distribución objetivo" en "Componentes de hardware".
- **Frontend (LNet):** Los clientes de Lustre y los nodos de servidor OSS/MDS se comunican a través de LNet

utilizando el tipo de red `@o2ib` con la pila OFED de NVIDIA. InfiniBand y RoCE son ambos transportes frontend validados. La colección ansible-lustre configura todas las interfaces frontend para LNet multivía, que agrega varios puertos para aumentar el ancho de banda y proporcionar redundancia de rutas para el tráfico de clientes y entre nodos.

- **Gestión y clúster:** Una red de gestión fuera de banda permite la gestión de servidores, el acceso a BMC y la gestión de matrices. Los agentes de fencing STONITH, como `fence_redfish`, usan esta red para llegar a los BMC de los servidores y cortar la alimentación de un nodo fallido. Corosync también puede ejecutar la comunicación del clúster sobre esta red como un anillo adicional junto con la estructura frontend. Configurar Corosync con varios anillos añade redundancia a la mensajería del clúster y reduce el riesgo de que una sola falla de red interrumpa el quórum.

### Clientes de Lustre

Un cliente de Lustre carga el módulo del kernel del cliente, establece la conectividad LNet con los nodos del servidor OSS/MDS y monta el sistema de archivos para presentar a las aplicaciones un único espacio de nombres coherente y compatible con POSIX. La E/S se distribuye en paralelo entre los OST activos. Los nodos cliente son externos a un bloque de construcción y usan el sistema de archivos a través de la red LNet de frontend. Los sistemas operativos de los clientes no aparecen en IMT para esta solución; usa el "Matriz de compatibilidad de Lustre" para ver las distribuciones de cliente probadas y las versiones de kernel compatibles con la versión de Lustre implementada (por ejemplo, Red Hat Enterprise Linux 9, SUSE Linux Enterprise Server 15 y Ubuntu 24.04).

Los nodos cliente deben tener conectividad con la misma estructura LNet y el mismo tipo de red que los nodos del servidor OSS/MDS. Si es necesario, un router LNet puede conectar subredes o estructuras LNet independientes.

NetApp ofrece paquetes RPM precompilados del servidor Lustre para Rocky Linux 9.8 y Red Hat Enterprise Linux 9.8. NetApp no ofrece paquetes de cliente precompilados. Compila los paquetes de cliente para el sistema operativo y el kernel a partir del código fuente de Lustre en el "Repositorio netapp-lustre".

Después de que se instalan los paquetes de cliente, la función opcional `lustre_client` en "colección ansible-lustre" configura las interfaces de red del cliente y LNet, habilita LNet multirrail cuando se selecciona, valida la conectividad y gestiona una unidad de montaje persistente de systemd. La función no compila Lustre. Instala los paquetes de cliente solo cuando `lustre_client_packages` está rellenado. Si quieres, puedes configurar los clientes manualmente. Para ver los procedimientos paso a paso, consulta "Implementa la solución" y el "documentación del rol lustre_client".

# Lustre con NetApp E-Series Storage: componentes de hardware

Utiliza estos requisitos de hardware para servidores, E-Series de NetApp, distribución de almacenamiento y redes cuando dimensionas y solicitas Lustre con NetApp E-Series Storage. Para los componentes de software, consulta "Componentes de software".

### Requisitos del servidor

La solución utiliza un modelo «trae tu propio servidor» (BYOS). Cada bloque de construcción requiere dos nodos de servidor OSS/MDS que cumplan o superen las siguientes especificaciones.

### Especificaciones de los nodos

La siguiente tabla muestra los requisitos mínimos del servidor por cada nodo OSS/MDS.

| Componente | Requisito |
|---|---|
| Cantidad | 2 nodos por building block |
| CPU | AMD EPYC o Intel Xeon, 32 núcleos o más |
| Memoria | 256 GB DDR5 (mínimo) |
| HCA de red | 6 HCA de 200 Gb de doble puerto (véase Conectividad HCA) |
| Ranuras PCIe | 2× PCIe Gen5 x16 y 4× PCIe Gen5 x8 (véase Requisitos de las ranuras PCIe) |
| Unidades de arranque | 2 unidades en RAID 1 (se recomienda RAID por software o por hardware) |

NetApp validó la solución usando servidores Lenovo ThinkSystem SR665 V3. Se admiten servidores equivalentes de otros fabricantes siempre que cumplan estos requisitos.

### Conectividad HCA

En la siguiente tabla se detallan los requisitos de HCA, puerto frontend LNet y ruta NVMe-oF por cada nodo de servidor OSS/MDS.

| HCA por nodo | Puertos LNet por nodo Lustre | Rutas NVMe-oF por nodo Lustre | Ejemplo HCA |
|---|---|---|---|
| 6 (12 puertos) | 4 | 8 en total (4 en cada cabina) | MCX755106AS-HEAT (tarjeta PCIe Gen5 de puerto dual de 200 Gb) |

NetApp recomienda seis HCA por nodo para maximizar el ancho de banda de las cabinas de almacenamiento EF80.

### Requisitos de las ranuras PCIe

Cada nodo de servidor OSS/MDS de un módulo EF80 alberga seis HCA de puerto doble: dos para LNet y cuatro para NVMe-oF. El ancho de la ranura determina el ancho de banda disponible para cada HCA, así que confirma el ancho eléctrico de cada ranura y cada riser en vez de solo el ancho de la tarjeta. Una tarjeta Gen5 x16 instalada en una ranura Gen5 x8 funciona a x8.

- **HCA de LNet:** Instálalos en ranuras PCIe Gen5 x16 para alcanzar el rendimiento máximo del frontend.
- **Adaptadores HCA NVMe-oF:** Instálalos en ranuras PCIe Gen5 x8 o PCIe Gen5 x16.
- **Equilibrio NUMA:** Distribuye los HCA de manera uniforme entre las dos zonas NUMA, de modo que cada zona albergue dos interfaces LNet y cuatro interfaces NVMe-oF.

La siguiente tabla muestra las ranuras mínimas para cada nodo de servidor OSS/MDS en un bloque de construcción EF80.

| Tipo de tráfico | HCA por nodo | Ancho mínimo de la ranura | Ranuras por zona NUMA |
|---|---|---|---|
| LNet | 2 | PCIe Gen5 x16 | 1 |
| NVMe-oF | 4 | PCIe Gen5 x8 | 2 |

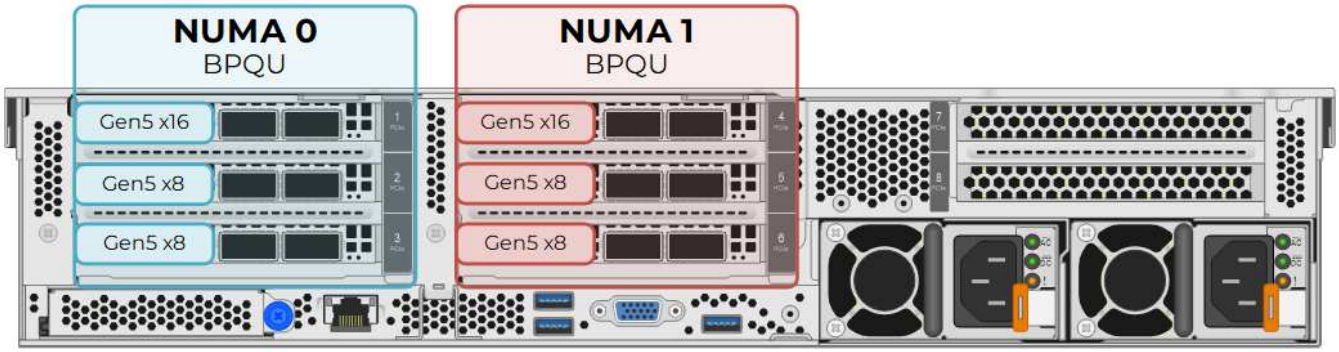
Si lo deseas, puedes dividir los puertos de un HCA de doble puerto para que un puerto transporte el tráfico de LNet y el otro transporte el tráfico de NVMe-oF. Instala los cuatro HCA en ranuras PCIe Gen5 x16 para que cada puerto LNet alcance su rendimiento máximo, y luego añade dos HCA más en ranuras Gen5 x8 o Gen5 x16 para los puertos NVMe-oF restantes.

▼ **Ejemplos de configuraciones de elevadores para Lenovo ThinkSystem SR665 V3**

Ambas combinaciones de tubos ascendentes siguientes cumplen los requisitos de ranuras para un bloque de construcción EF80.

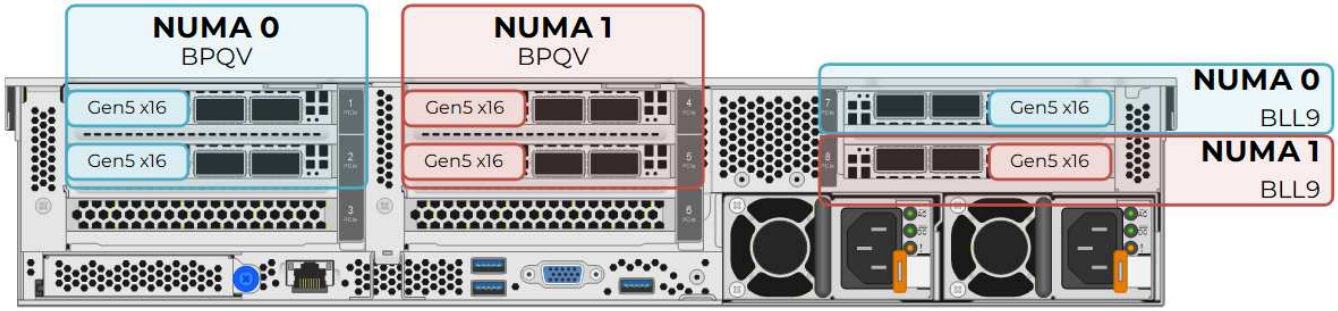
**Anchos mínimos de ranura**

Instala un riser BPQU en las posiciones 1 y 2. Cada riser BPQU ofrece una ranura PCIe Gen5 x16 y dos ranuras PCIe Gen5 x8. En conjunto, los risers proporcionan dos ranuras Gen5 x16 para LNet y cuatro ranuras Gen5 x8 para NVMe-oF, repartidas equitativamente entre las zonas NUMA.




**Todas las ranuras x16 de Gen5**

Instala un riser BPQV en las posiciones de riser 1 y 2, y un riser BLL9 en la posición de riser 3. Cada riser BPQV proporciona dos ranuras PCIe Gen5 x16. El riser BLL9 proporciona la ranura 7 en la zona NUMA 0 y la ranura 8 en la zona NUMA 1, ambas PCIe Gen5 x16. Esta combinación proporciona seis ranuras Gen5 x16, tres por zona NUMA, y permite dividir el tráfico LNet y NVMe-oF entre los puertos del mismo HCA.




## Bloque de construcción estándar EF80

La EF80 es la plataforma estándar validada para esta versión de la solución.

### Especificaciones de la cabina

La siguiente tabla muestra las especificaciones de la cabina EF80 para un bloque básico (dos cabinas).

| Componente | Especificación |
|---|---|
| Modelo | NetApp EF80 |
| Factor de forma | Chasis base de 2U, 24 ranuras internas para SSD NVMe |
| Controladoras | Controladoras dobles (A y B) |
| Unidades | 24 × SSD NVMe por matriz |
| Conectividad de E/S | 8× 200 Gb NVMe/IB o NVMe/RoCE puertos de host por cabina en el diseño validado de Lustre |

La plataforma EF80 admite hasta doce puertos de host por array cuando se instalan tres módulos de E/S de host de dos puertos en cada controlador. El diseño validado de Lustre utiliza módulos de E/S de host en las ranuras 1 y 2 para ocho puertos de host por array.

Cada cabina EF80 también utiliza el módulo de E/S dedicado de la ranura 4 para la duplicación entre controladores. Conecta el puerto 4a del controlador A al puerto 4a del controlador B, y el puerto 4b del controlador A al puerto 4b del controlador B. Estas conexiones proporcionan duplicación de caché y envío de E/S, y no se usan para el tráfico NVMe-oF del host. Consulta "Conecta los cables de las conexiones de duplicación entre los controladores EF50 y EF80".

Para consultar las especificaciones completas de la EF-Series en todos los modelos, consulta la "NetApp Ficha técnica de la cabina all-flash EF-Series".

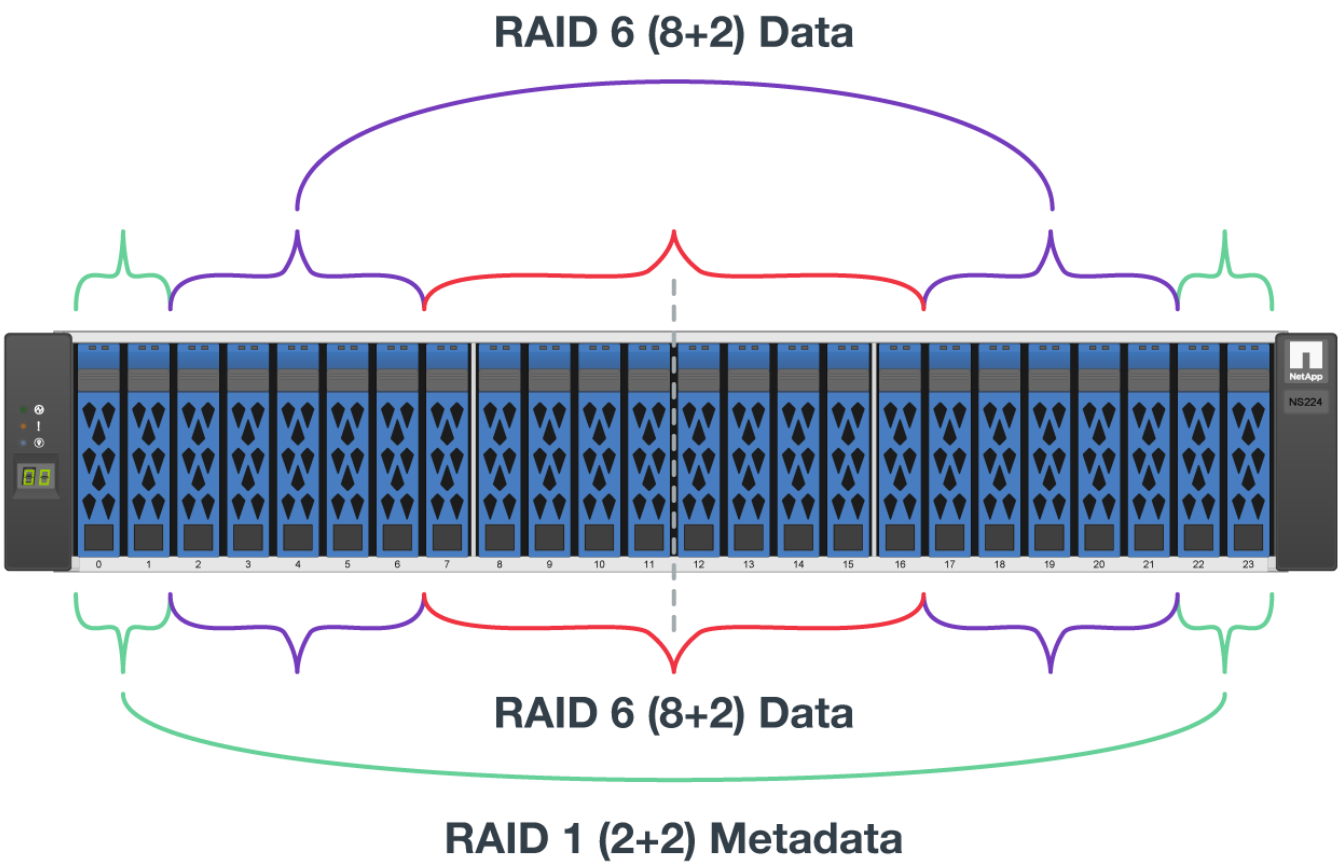
### Distribución de unidades (24 unidades por cabina)

Cada cabina EF80 de un bloque básico utiliza las veinticuatro ranuras para unidades NVMe:

- 4 unidades en RAID 1 para almacenamiento MGS/MDT (grupo de volúmenes compartido o asignación DDP)
- 10 unidades en RAID 6 para almacenamiento OST (primer grupo OST)
- 10 unidades en RAID 6 para almacenamiento OST (segundo pool OST)

Esta distribución se aplica cuando usas unidades de 3,84 TB, 7,68 TB o 15,3 TB con grupos de volúmenes tradicionales. Cuando uses unidades Capacity Flash (QLC) de 30,7 TB o 61,4 TB, aprovisiona un solo Dynamic Disk Pool en las veinticuatro unidades y crea volúmenes RAID 1 dentro del pool para los volúmenes MGS y MDT, mientras que para los OST usa volúmenes RAID 6 predeterminados.

> Actualmente no se recomiendan las unidades de 1,92 TB para esta solución. Usa una de las capacidades de unidad validadas que se indican arriba.




**Recuento de volúmenes y objetivos (elemento básico)**

La siguiente tabla muestra los recuentos de MGS, MDT y OST para un bloque de construcción básico.

| Tipo de objetivo | Recuento por BB | RAID / pool | Notas |
|---|---|---|---|
| MGS | 1 | RAID 1 | Solo bloque básico; array 1 |
| MDT | 8 | RAID 1 | 4 por matriz |
| OST | 32 | RAID 6 (TLC) o DDP (QLC) | 16 por matriz |

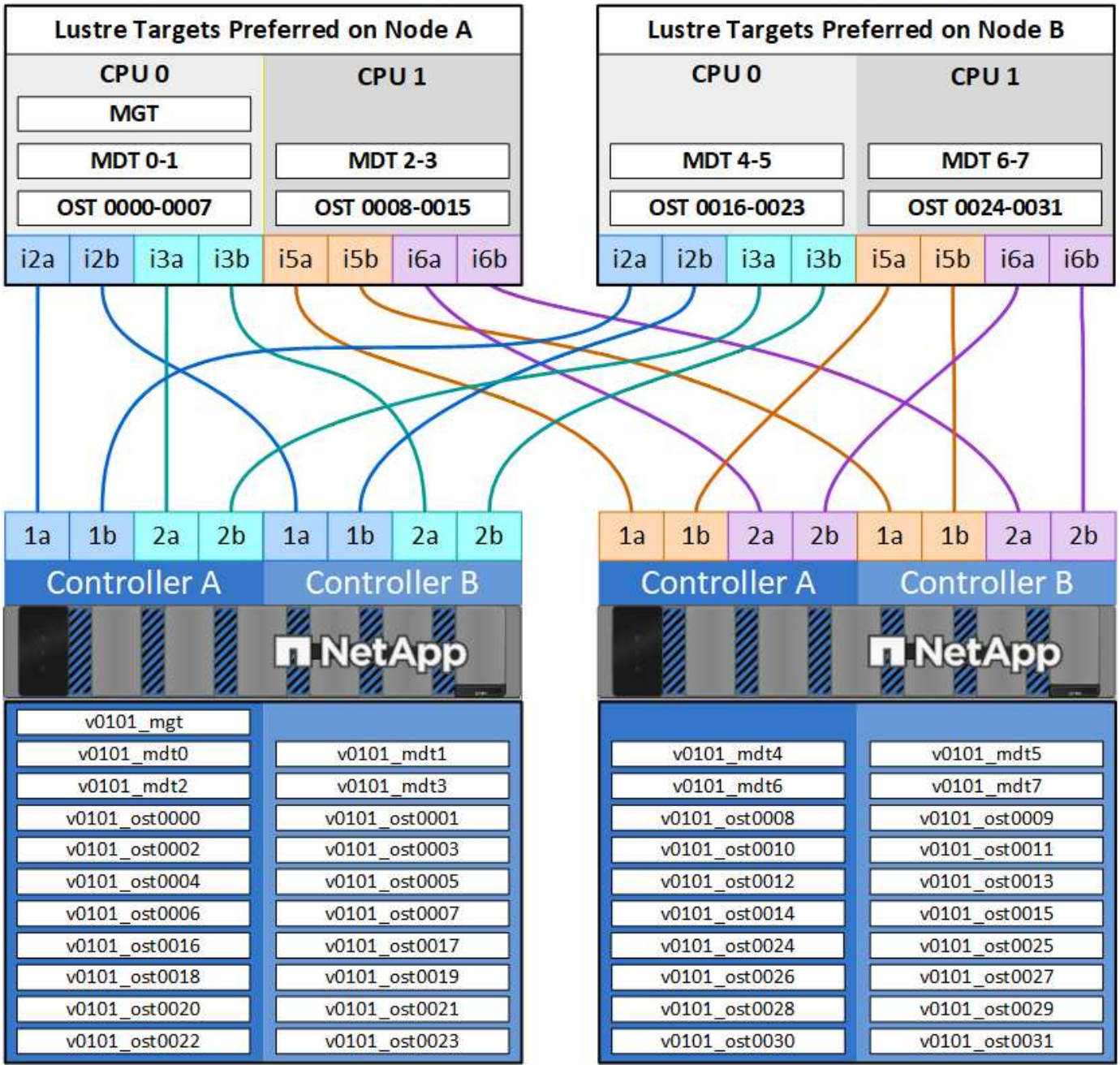
Utiliza las siguientes directrices para el ajuste de tamaño de volúmenes como punto de partida en el inventario de Ansible.

| Volumen | Tamaño recomendado | Notas |
|---|---|---|
| MGS | 5–10 GiB | Solo datos de configuración |
| MDT | Capacidad de RAID 1 ÷ número de MDT | Aproximadamente entre 1 y 2 TiB cada uno (valores típicos) |
| OST | Capacidad RAID 6 o DDP ÷ número de OST por grupo | Escala con la capacidad de la unidad; consulta "Guía de tallas" |

### Distribución del volumen principal de bloques de construcción

La siguiente figura muestra la ubicación recomendada de los destinos Lustre y la conectividad NVMe-oF entre los nodos de servidor OSS/MDS y las matrices E-Series en un bloque de construcción EF80 básico. En el diagrama, el destino de gestión se indica como **MGT** (destino de gestión); un MGT aloja el servicio MGS (servidor de gestión) al que se hace referencia en otras partes de este documento.

#### Distribución de destino de los bloques básicos (EF80)




La siguiente tabla muestra cómo se distribuyen los volúmenes entre las dos matrices y qué servidor prefiere cada destino.

| Tipo de objetivo | Matriz 1 | Matriz 2 | Servidor 1 | Servidor 2 | Notas |
|---|---|---|---|---|---|
| MGS | 1 | 0 | 1 | 0 | Solo el bloque básico |

| Tipo de objetivo | Matriz 1 | Matriz 2 | Servidor 1 | Servidor 2 | Notas |
|---|---|---|---|---|---|
| MDT | 4 | 4 | 4 | 4 | Cada servidor prefiere 2 MDT de cada array |
| OST | 16 | 16 | 16 | 16 | 8 por grupo RAID 6 × 2 grupos por array, o una asignación equivalente de DDP |
| **Volúmenes totales** | **21** | **20** | **21** | **20** | |

## Selección de grupos de almacenamiento: TLC frente a QLC

Elige el tipo de grupo de E-Series en función de la capacidad de las unidades NVMe de las matrices:

- **Unidades de 3,84 TB, 7,68 TB y 15,3 TB:** Usa **grupos de volúmenes RAID 6** para los volúmenes OST y **grupos de volúmenes RAID 1** para los volúmenes MGS y MDT, o usa DDP para el conjunto de unidades compartidas.
- **Unidades flash (QLC) con una capacidad de 30,7 TB y 61,4 TB:** Utiliza únicamente **Dynamic Disk Pools (DDP)**. Crea volúmenes RAID 1 dentro del DDP para el almacenamiento de MGS y MDT. Crea volúmenes RAID 6 para el almacenamiento de OST a partir del DDP.

Para consultar estimaciones de la capacidad utilizable por módulo según el tamaño y la disposición de la unidad, consulta "Guía de tallas".

## Requisitos de red

### Backend (NVMe-oF)

- NVMe/InfiniBand o NVMe/RoCE entre cada nodo OSS/MDS y cada matriz E-Series
- MTU 9000 en interfaces de backend NVMe/RoCE (típico)
- Ocho rutas desde cada nodo Lustre hasta las matrices de almacenamiento (cuatro hacia cada matriz)
- EF80 utiliza seis HCA por nodo (i2, i3, i5 e i6 para NVMe-oF; i1 e i4 para LNet). Conecta el nodo A a los puertos del controlador cuyas etiquetas terminen en `a`, como `1a` y `2a`. Conecta el nodo B a los puertos del controlador cuyas etiquetas terminen en `b`, como `1b` y `2b`. Consulta "Cableado del backend EF80 de seis HCA".

### Frontend (LNet)

- ¿IPoIB o RoCE para LNet (`@o2ib` tipo de red?
- MTU 9000 en las interfaces frontend de RoCE (típico)
- Se recomienda utilizar LNet multirrail cuando se disponga de cuatro puertos front-end (EF80: i1 e i4 HCAs, ambos puertos)
- Conmutador dedicado InfiniBand o RoCE que conecta los nodos del servidor OSS/MDS y los clientes Lustre
- Se recomienda RoCE sin pérdidas (PFC) en fabrics RoCE

**Gestión**

- Red de gestión fuera de banda para los BMC de los servidores y los puertos de gestión de array
- Red Corosync dedicada o estructura de red de frontend compartida (dependiendo del sitio)

# Lustre con almacenamiento E-Series de NetApp - componentes de software

Revisa la pila de software y la automatización mediante Ansible que implementa Lustre con el almacenamiento E-Series de NetApp. Para servidores, almacenamiento y redes, consulta "Componentes de hardware".

## Software de bloques de construcción

En la siguiente tabla se enumeran los componentes de software utilizados por las matrices E-Series y los nodos de servidor OSS/MDS. Usa el "Herramienta de matriz de interoperabilidad (IMT) de NetApp" como fuente autorizada para las versiones compatibles y combinaciones de componentes. Busca **E-Series Lustre** y confirma el sistema operativo, Lustre, SANtricity OS, DOCA-OFED, el firmware del HCA y las versiones del paquete HA antes de la implementación.

| Componente | Función |
|---|---|
| SO SANtricity | Se ejecuta en cada matriz E-Series y ofrece SANtricity System Manager, alta disponibilidad con controladores dual-activos y conmutación automática de rutas de E/S en caso de fallo, equilibrio de carga automático y supervisión de rutas, Dynamic Disk Pools y RAID tradicional, reconstrucciones automáticas de unidades, protección de caché duplicada, Data Assurance, supervisión proactiva del estado de las unidades y cambios en línea de software, firmware y configuración. |
| Lustre | Ofrece a los clientes un único espacio de nombres compatible con POSIX, al tiempo que distribuye los metadatos y los datos de los archivos entre varios destinos para permitir el acceso paralelo. El servidor de gestión (MGS) almacena la configuración del sistema de archivos, los destinos de metadatos (MDT) gestionan el espacio de nombres y los destinos de almacenamiento de objetos (OST) almacenan los datos de los archivos en volúmenes E-Series. Esta separación permite escalar el rendimiento y la capacidad añadiendo bloques de construcción. |
| Red Hat Enterprise Linux (RHEL) o Rocky Linux | Proporciona el núcleo, los servicios del sistema y el marco de paquetes para los servicios de destino de Lustre, la agrupación en clústeres de alta disponibilidad (HA) y la conectividad de almacenamiento y red. Los repositorios de los sistemas operativos compatibles también proporcionan Pacemaker, Corosync, agentes de fence, NVMe-oF y paquetes de multivía, que se han probado juntos como una pila de soluciones. |
| Servicios de clúster de alta disponibilidad (Pacemaker, Corosync y fence agents) | En conjunto, estos servicios proporcionan una alta disponibilidad coordinada para los destinos de Lustre en todos los nodos de servidor OSS/MDS. Mantienen la pertenencia al clúster y el quórum, supervisan los recursos, orquestan la conmutación por error ordenada y aíslan los nodos que no responden antes de activar destinos compartidos en otros lugares. Esta coordinación evita el acceso de split-brain a los volúmenes E-Series y protege la integridad del sistema de archivos durante los fallos. |

| Componente | Función |
| --- | --- |
| NVIDIA DOCA-OFED | Proporciona controladores HCA, bibliotecas RDMA y utilidades de gestión para InfiniBand y estructuras RoCE. La misma pila de software admite el acceso NVMe-oF de baja latencia al backend de las cabinas E-Series y la comunicación LNet de alto rendimiento al frontend con los clientes Lustre. |

NetApp ofrece paquetes RPM de servidor Lustre precompilados para Rocky Linux 9.8 y Red Hat Enterprise Linux 9.8.

## Automatización con Ansible

Lustre con el almacenamiento E-Series de NetApp se suministra y se implementa mediante la automatización de Ansible desde un nodo de control con acceso de gestión a las matrices E-Series y a los nodos de servidor OSS/MDS. Un inventario de Ansible modela la solución deseada, incluyendo matrices, servidores, asignaciones de almacenamiento, interfaces de red y recursos del clúster de alta disponibilidad (HA).

La colección Ansible NetApp E-Series Lustre orquesta la implementación de principio a fin usando las colecciones SANtricity y Host. Juntas, estas colecciones aprovisionan y asignan el almacenamiento E-Series, preparan el sistema operativo del servidor, configuran la conectividad NVMe-oF y LNet, implementan los targets de Lustre y crean recursos Pacemaker. Esta automatización hace que la configuración sea repetible y simplifica agregar bloques de construcción a un sistema de archivos existente.

La siguiente tabla resume los principales paquetes de software utilizados en el nodo de control de Ansible. Consulta el "Colección Ansible de Lustre para E-Series de NetApp" para conocer las dependencias actuales de los paquetes.

| Paquete | Dependencia o finalidad |
| --- | --- |
| Pitón | Proporciona el entorno de ejecución de Ansible y los módulos de Python necesarios. |
| ansible-core | Requiere Python y ejecuta las colecciones de Ansible de E-Series de NetApp. |
| Paquetes adicionales de Python | `cryptography`, `ipaddr`, `netaddr`, `passlib`, `resolvelib`, `jinja2` y `pyyaml` |
| Colección Ansible de Lustre para E-Series de NetApp | Configura Lustre, LNet, la conectividad de host NVMe-oF y los recursos de Pacemaker. |
| Colección de Ansible de SANtricity para E-Series de NetApp | Configura las matrices E-Series, los pools de almacenamiento, los volúmenes y las asignaciones de host a través de la API de SANtricity Web Services. |
| Colección de Ansible para hosts de E-Series de NetApp | Prepara los nodos de servidor OSS/MDS para la conectividad LNet y NVMe-oF con las cabinas E-Series. |

## Software cliente de Lustre

El software cliente de Lustre permite a un sistema acceder al sistema de archivos paralelo a través de la estructura LNet de frontend. Cada cliente necesita los módulos del kernel de Lustre y las utilidades de cliente de Lustre que sean compatibles tanto con el kernel que está ejecutando como con la versión de Lustre implementada en los servidores.

Utiliza la "Matriz de compatibilidad de Lustre" para encontrar un sistema operativo cliente y un kernel

compatibles con tu versión de Lustre. NetApp no ofrece paquetes de cliente de Lustre precompilados. Compila los paquetes de cliente para el sistema operativo y el kernel del cliente a partir del código fuente de Lustre proporcionado por NetApp en el "Repositorio netapp-lustre". Después de instalar los paquetes de cliente de Lustre, el rol `lustre_client` puede configurar las interfaces de red, LNet y crear una unidad de montaje persistente de systemd para el sistema de archivos Lustre.

Para conocer los pasos de implementación en el cliente, consulta "Implementa la solución".

# Lustre con almacenamiento NetApp E-Series: guía de dimensionamiento

Adapta Lustre con los bloques de construcción de almacenamiento E-Series de NetApp según la capacidad y las necesidades de metadatos, decide cuándo añadir capacidad y revisa los factores que afectan al rendimiento.

## Ajuste de tamaño de la capacidad

Un módulo básico con dos matrices EF80 ofrece la siguiente disposición de destinos de Lustre. Para conocer la ubicación recomendada de los destinos y las rutas NVMe-oF, consulta "Distribución del volumen principal de bloques de construcción" en "Componentes de hardware".

| Componente | Recuento | Tamaño habitual del volumen (por objetivo) |
|---|---|---|
| MGS | 1 | 5-10 GiB |
| MDT | 8 | El tamaño varía en función de la capacidad de la unidad y de la configuración RAID |
| OST | 32 | El tamaño varía en función de la capacidad de la unidad y de la configuración RAID |

Cada matriz EF80 utiliza veinticuatro unidades NVMe. Las capacidades admitidas son 3,84 TB, 7,68 TB y 15,3 TB con grupos de volúmenes tradicionales (RAID 1 para MGS/MDT y RAID 6 para OST) o DDP, y unidades Capacity Flash (QLC) de 30,7 TB o 61,4 TB solo con DDP. Actualmente no se recomiendan las unidades de 1,92 TB para esta solución. Para conocer la disposición de las ranuras de unidad y la selección de grupos de unidades, consulta "Componentes de hardware".

En la siguiente tabla se indica la capacidad utilizable aproximada de un módulo EF80 (dos matrices, 48 unidades). La capacidad utilizable de la matriz no es la misma que la capacidad utilizable final del sistema de archivos Lustre. La asignación de nodos de información del MDT, los registros en diario, el sobreaprovisionamiento y los porcentajes de volumen de inventario reducen la capacidad disponible para las aplicaciones.

| Capacidad de la unidad | Disposición (por matriz) | Capacidad utilizable aproximada (módulo) |
|---|---|---|
| 3,84 TB | RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10) | 138,08 TB |
| 3,84 TB | DDP (24 unidades, 2 reservadas) | 133,63 TB |
| 7,68 TB | RAID 1 (2+2) + RAID 6 (10) + RAID 6 (10) | 276,35 TB |

| Capacidad de la unidad | Disposición (por matriz) | Capacidad utilizable aproximada (módulo) |
|---|---|---|
| 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 |

Define los metadatos y los datos por separado y, a continuación, configura los tamaños de los volúmenes en el inventario de Ansible. Las plantillas de implementación formatean los destinos con ldiskfs; MDT utiliza la proporción de nodos de información predeterminada de Lustre, y las plantillas OST establecen `-i 4096`.

- **Metadatos (MDT):** Dimensiona los MDT en función del número de archivos previsto. Calcula aproximadamente el doble del número de archivos previsto para tener margen de crecimiento. Un MDT de tamaño insuficiente puede impedir la creación de nuevos archivos incluso cuando aún quede capacidad en el OST.
- **Datos (OST):** Determina el tamaño de los OST a partir de la capacidad utilizable y las necesidades de rendimiento.
- **MGS:** Utiliza un objetivo de gestión pequeño (5-10 GiB) únicamente para la configuración del sistema de archivos.

Ajusta los tamaños de volumen MDT y OST en el inventario según la capacidad de la unidad seleccionada. Modifica `format_options.mkfsoptions` en `eseries_lustre_filesystem_mdt` o `eseries_lustre_filesystem_ost` solo cuando un sitio necesite una relación de nodos de información diferente. Para información sobre las relaciones de nodos de información, consulta "Lustre Wiki: Ajuste de Lustre". Para saber cuándo añadir bloques de construcción, consulta Escalado.

## Escalado

Añade módulos cuando la capacidad o la carga de metadatos lo requiera:

- **Solo OST:** El número de archivos y la carga de metadatos se encuentran dentro de la capacidad actual de MDT, y necesitas más capacidad de datos o un mayor rendimiento agregado. Agrega 32 OST y dos nodos de servidor OSS/MDS.
- **MDT+OST:** El número de archivos o la tasa de metadatos se está acercando a la capacidad del MDT, o necesitas tanto más rendimiento de metadatos como más capacidad de datos. Añade 8 MDT y 32 OST.

Utiliza `mdt.*.md_stats` en los MDT existentes para comprobar si los metadatos están saturados. Limita cada clúster de Pacemaker/Corosync a cinco bloques de construcción (diez nodos de servidor OSS/MDS). Para implementaciones más grandes, distribuye los bloques de construcción de manera uniforme entre varios clústeres de alta disponibilidad (HA). Consulta "Arquitectura de la solución" para conocer las reglas de escalabilidad.

El siguiente ejemplo muestra un sistema de archivos con un bloque básico y un bloque exclusivo de OST en un clúster HA.

| Bloque de construcción | Tipo | Servidores | Matrices | MGS | MDT | OST |
|---|---|---|---|---|---|---|
| BB1 | Base | 2 | 2 | 1 | 8 | 32 |
| BB2 | Solo OST | 2 | 2 | 0 | 0 | 32 |
| Total | | 4 | 4 | 1 | 8 | 64 |

BB2 registra sus objetivos OST contra el MGS en BB1 usando `mgsnode=`. Los índices OST de BB2 se asignan globalmente (por ejemplo, OST0032 hasta OST0063) para que ningún índice entre en conflicto con BB1.

### Actuación

La validación formal utiliza IOR, mdtest y fio desde clientes Lustre para caracterizar el rendimiento relativo, las IOPS y el comportamiento de los metadatos, y para validar la conmutación por error de HA. Estas pruebas no publican cifras de rendimiento garantizadas. Las pruebas sintéticas miden el comportamiento en el mejor de los casos y pueden no reflejar el rendimiento real de las aplicaciones.

El rendimiento escala aproximadamente con el número de bloques de construcción. Los resultados obtenidos dependen del tipo de unidad y de la configuración del grupo, del número de rutas NVMe-oF, de la estructura LNet y del número de clientes, así como de la proporción entre E/S de datos y E/S de metadatos.

Ponte en contacto con el equipo de tu cuenta de NetApp para obtener recomendaciones sobre el dimensionamiento y el rendimiento específicas para tu carga de trabajo.

Para ver la configuración de las unidades y el número de volúmenes, consulta "Componentes de hardware". Para ver los pasos de implementación, consulta "Implementa la solución". Para ver la guía de inventario a nivel de campo, consulta "Personaliza el inventario de Ansible de Lustre".

# Implementa la solución

## Implementa Lustre con el almacenamiento E-Series de NetApp

Implementa Lustre con NetApp E-Series Storage completando en orden las tareas de hardware, preparación del entorno, inventario, implementación de Ansible y verificación.

#### Prerrequisitos

Antes de iniciar la implementación, asegúrate de lo siguiente:

- Has seguido las instrucciones sobre ajuste de tamaño de volúmenes y módulos de "Guía de tallas" y has seleccionado hardware que coincide con "Componentes de hardware".
- Has confirmado que la combinación de servidor, HCA, E-Series, sistema operativo, Lustre, SANtricity OS y DOCA-OFED figura en la lista de **E-Series Lustre** en "Herramienta NetApp Interoperability Matrix Tool".
- Todos los nodos de servidor OSS/MDS y las cabinas E-Series para los módulos previstos ya están en las instalaciones y listos para montar en rack.
- Dispones de credenciales y acceso a la red para los BMC de los servidores, los puertos de gestión de las cabinas y el host que hará las veces de nodo de control de Ansible.

NetApp admite implementar el clúster HA de Lustre y los clientes de Lustre con el "Colección

> Ansible de Lustre para E-Series de NetApp". No configures manualmente los destinos de Lustre, LNet, la conectividad de host NVMe-oF ni los recursos de Pacemaker.

**Flujo de trabajo de implementación**

Realiza las siguientes tareas en el orden indicado:

1. "Rack y cableado de hardware Lustre".

   Instala los servidores Lustre y las cabinas EF80, y luego conecta las estructuras NVMe-oF del backend y LNet del frontend.

2. "Prepara el entorno de implementación de Lustre".

   Configura las cabinas y los servidores, luego instala Ansible y sus dependencias en el nodo de control.

3. "Personaliza el inventario de Ansible de Lustre".

   Selecciona la plantilla para la estructura del sitio y configura los servidores, las cabinas, las redes, los componentes básicos y las credenciales.

4. "Implementa el clúster Lustre de alta disponibilidad y los clientes".

   Prepara el sistema operativo del servidor, instala NVIDIA DOCA-OFED, ejecuta el playbook de implementación principal y configura los clientes de Lustre.

5. "Verifica y amplía la implementación de Lustre".

   Comprueba el estado del clúster, los montajes y la capacidad antes de su uso en producción. Amplía el inventario existente cuando el sistema de archivos necesite otro bloque de construcción.

**Información relacionada**

- "Colección Ansible de Lustre para E-Series de NetApp"
- "NetApp repositorio de código fuente de Lustre"
- "Guía de usuario de la administración de HA"
- "Guía de ajuste de rendimiento"
- "Plantillas de implementación de clientes de Lustre"

## Rack y cableado de hardware Lustre

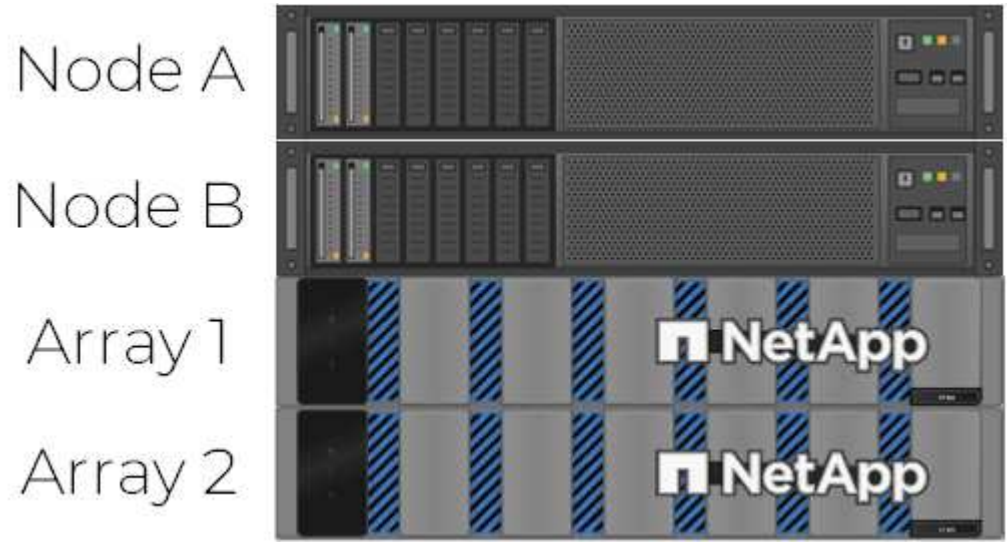
Instala los servidores Lustre y las matrices EF80 de NetApp, y luego conecta las estructuras NVMe-oF del backend y LNet del frontend para cada building block.

**Antes de empezar**

Consulta los recuentos de HCA, los requisitos de ranuras PCIe, los requisitos de ruta y las directrices sobre MTU en "Componentes de hardware".

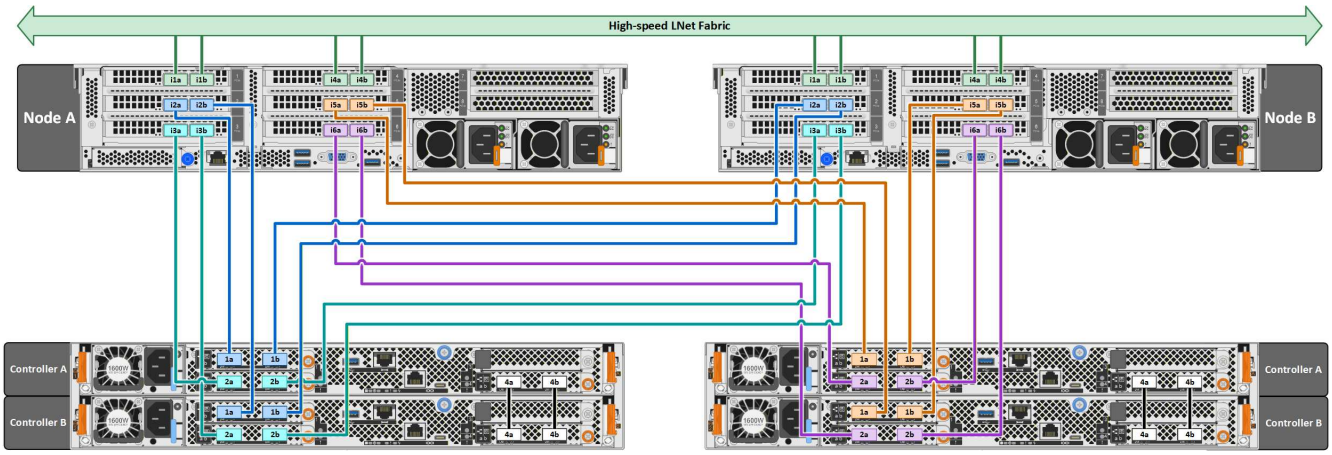
**Pasos**

1. Instala en rack y conecta todos los servidores Lustre y las matrices de almacenamiento E-Series según la topología de bloques modulares.




2. Para cada matriz EF80, conecta el puerto 4a del controlador A al puerto 4a del controlador B y el puerto 4b del controlador A al puerto 4b del controlador B. Estas conexiones dedicadas permiten la duplicación entre controladores y no se usan para el tráfico del host. Consulta "Conecta los cables de las conexiones de duplicación entre los controladores EF50 y EF80".
3. Conecta la red NVMe-oF de backend entre los servidores y las arrays según la topología de seis HCA de EF80.

**Cableado trasero EF80 de seis HCA (RoCE o InfiniBand)**




4. Conecta las cuatro interfaces LNet frontales (los HCA i1 e i4) de cada servidor Lustre y de los clientes Lustre a la estructura de red dedicada de InfiniBand o RoCE.
5. Al utilizar RoCE, configura la estructura de red para que funcione sin pérdidas con control de flujo prioritario (PFC). Los playbooks de Ansible configuran RoCE sin pérdidas en todas las interfaces del building block.

**Después de que termines**

"Prepara los servidores, las matrices y el nodo de control de Ansible".

## Prepara el entorno de implementación de Lustre

Configura el acceso de administración y los ajustes básicos en cada matriz EF80 y servidor Lustre, luego prepara un nodo de control de Ansible para implementar la

solución.

### Prepara las cabinas E-Series

**Pasos**

1. Configura la interfaz de gestión en cada controlador y comprueba que SANtricity System Manager sea accesible.
2. Establece el nombre de la matriz y las credenciales de administrador que utilizará el inventario de Ansible.
3. Comprueba que la matriz esté en un estado óptimo y que todas las unidades estén presentes y reporten correctamente.

Para obtener instrucciones sobre cómo configurar la conexión de gestión, consulta la "Documentación de E-Series".

### Prepara los servidores Lustre

**Pasos**

1. Configura el controlador de gestión de la placa base (BMC) en cada servidor para el acceso fuera de banda y ajusta la configuración del sistema UEFI para obtener el máximo rendimiento.
2. Instala un sistema operativo compatible (RHEL o Rocky Linux), luego configura el nombre de host y el puerto de red de gestión.
3. Prepara todos los paquetes de firmware y software para que cumplan los requisitos de la solución **E-Series Lustre**, tal y como se describe en "Componentes de software".

### Prepara el nodo de control de Ansible

Designa un sistema Linux, ya sea virtual o físico, con acceso a la red de gestión de todos los servidores Lustre y las matrices E-Series, para que actúe como nodo de control de Ansible.

Los siguientes pasos se refieren a Ubuntu 24.04. Para obtener instrucciones sobre otra distribución de Linux, consulta "Documentación de instalación de Ansible".

**Pasos**

1. Instala Python, pip y el paquete de entorno virtual de Python:

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

2. Crea y activa un entorno virtual de Python:

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

3. Instala Ansible y los paquetes de Python necesarios en el entorno virtual activado:

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

4. Instala la colección Ansible NetApp E-Series Lustre. La instalación de la colección también instala las colecciones de las que depende, incluidas las colecciones SANtricity y Host:

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

5. Comprueba las versiones instaladas de Ansible, Python y la colección:

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

6. Configura el acceso SSH sin contraseña desde el nodo de control a cada servidor Lustre. Sustituye `<ip_or_hostname>` por la dirección IP de gestión o el nombre de host de un servidor:

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

Repite `ssh-copy-id` para cada servidor Lustre.

> No configures el acceso SSH sin contraseña en las matrices E-Series. Ansible gestiona las matrices a través de la API de servicios web de SANtricity utilizando las credenciales definidas en el inventario.

**Después de que termines**

"Personaliza el inventario de Ansible".

## Personaliza el inventario de Ansible de Lustre

Crea una copia de trabajo de una plantilla de implementación de EF80 y personaliza su inventario para tus servidores, arrays, redes y componentes básicos de Lustre.

Mantén el inventario seleccionado en su directorio de estructura de red y plataforma. Conserva el directorio compartido `playbooks` y el archivo `passwords.yml` en la raíz de `building_blocks`.

### Selecciona una plantilla de implementación

Empieza por la plantilla que se ajuste a la estructura de red del sitio:

- **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 los ajustes específicos del sitio

Configura los valores específicos del sitio en el inventario, junto con los ajustes del repositorio local y de las reglas de udev. Las referencias de línea de la tabla apuntan a los archivos de plantilla RoCE EF80 en la

etiqueta `ansible-lustre release/1.0.0`. La plantilla InfiniBand utiliza los mismos nombres de variables y la misma estructura, salvo la variable de interfaz LNet, que se indica en la tabla.

| Campo | Variable de Ansible | Archivo de plantilla (`release/1.0.0`) | Ámbito | Notas |
|---|---|---|---|---|
| Nombre del sistema de archivos | `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" | A nivel de clúster | Anidado bajo `format_options` para los mapas MGT, MDT y OST. Máximo 8 caracteres. Usa el mismo valor en las tres ubicaciones. Una clave de nivel superior `fsname` no tiene efecto. |
| NIDs de MGS para el registro de objetivos | `eseries_lustre_filesystem_mdt.format_options.mgsnode`, `eseries_lustre_filesystem_ost.format_options.mgsnode` | "ha_cluster.yml#L98", "#L118" | A nivel de clúster | Anidados bajo MDT y OST `format_options`. Incluye los NID de LNet del frontend de ambos nodos del par de alta disponibilidad (HA). Enumera primero los NID del nodo configurado como propietario preferido del MGS, seguidos de los NID del nodo asociado. |
| Nombre del clúster de Pacemaker | `eseries_lustre_pacemaker_cib_properties_overrides.cluster_properties.cluster-name` | "ha_cluster.yml#L228" | A nivel de clúster | Anidado bajo las modificaciones de propiedades de Pacemaker CIB. Nombre único para el clúster de alta disponibilidad. |
| Direcciones de fencing de BMC | `eseries_lustre_pacemaker_fencing_agents.fence_redfish.ip` | "ha_cluster.yml#L211", "#L214", "#L217", "#L220" | Repite por cada servidor Lustre | Dirección IP de Redfish BMC para cada nodo del servidor. |
| Destinatarios de correo electrónico de alerta | `eseries_lustre_alert_email_list` | "ha_cluster.yml#L233" | A nivel de clúster | Destinatarios para las notificaciones de recursos y fallos de HA. |
| Dominio de correo electrónico | `eseries_lustre_mail_service_overrides.conf.mydomain` | "ha_cluster.yml#L240" | A nivel de clúster | Dominio de Postfix usado para enviar correos electrónicos de alerta. |

| Campo | Variable de Ansible | Archivo de plantilla (`release/1.0.0`) | Ámbito | Notas |
|---|---|---|---|---|
| Fuentes de tiempo NTP | `eseries_lustre_time_sync_overrides.server_pools` | "ha_cluster.yml#L250" | A nivel de clúster | Fuentes de tiempo para los nodos del clúster. Especifica un grupo de servidores NTP o uno o varios servidores NTP individuales. Incluye la directiva obligatoria `pool` o `server` y opciones como `iburst`. |
| Mapa de udev de interfaz PCI a nombre | `eseries_ip_udev_rules` | "ha_cluster.yml#L172" | Repite según la disposición de las ranuras HCA | Asigna a cada ranura PCI un nombre de interfaz persistente (valores en la línea 182). Recopílalos con `lspci -nnvmm`. |
| Repositorio local de paquetes (opcional) | `eseries_common_yum_repos` | "ha_cluster.yml#L256", "#L261" | A nivel de clúster | Para instalaciones aisladas físicamente, descomenta el bloque y configura el repositorio `url` (L264). |
| Dirección IP de administración del servidor | `ansible_host` | "host_vars/lustre_01.yml#L10" | Repite por cada servidor Lustre | Un archivo `host_vars/lustre_NN.yml` por servidor. |
| Interfaces de backend NVMe-oF | `eseries_nvme_roce_interfaces` (RoCE) / `eseries_nvme_ib_interfaces` (InfiniBand) | "RoCE host_vars/lustre_01.yml#L21", "InfiniBand host_vars/lustre_01.yml#L17" | Repite por cada servidor Lustre | Direcciones de interfaz de backend utilizadas para el tráfico de almacenamiento NVMe-oF hacia las matrices E-Series. |
| Interfaces de clúster de LNet | `eseries_roce_interfaces` (RoCE) / `eseries_ipoib_interfaces` (InfiniBand) | "RoCE host_vars/lustre_01.yml#L47", "InfiniBand host_vars/lustre_01.yml#L38" | Repite por cada servidor Lustre | Direcciones de la interfaz LNet del frontend (valores RoCE en L56; valores InfiniBand en L47). |
| Direcciones IP de los nodos de Corosync | `eseries_lustre_corosync_node_ips` | "host_vars/lustre_01.yml#L58" | Repite por cada servidor Lustre | Enumera todas las direcciones IP de los nodos en el clúster HA (valores en L62). |
| NID de los servicios de destino de Lustre | `lustre_target_service_nodes` | "host_vars/lustre_01.yml#L64" | Repite por cada servidor Lustre | Incluye los NID de LNet del frontend (`@o2ib`) de ambos nodos del par HA para que los servicios de destino sigan siendo accesibles tras una conmutación por error (valores en L67). |
| Dirección IP de gestión de matrices | `ansible_host` | "host_vars/netapp_01.yml#L13" | Repite por cada matriz E-Series | Dirección IP de gestión de un controlador; la URL de la API se deriva de ella. |

| Campo | Variable de Ansible | Archivo de plantilla (`release/1.0.0`) | Ámbito | Notas |
|---|---|---|---|---|
| Pertenencia a un grupo de bloques de construcción | `mgs` / `mds_NN` / `oss_NN` | "lustre_inventory.yml" | Por bloque de construcción | Cada nombre de host del inventario debe coincidir con el nombre base de su archivo `host_vars` correspondiente. Por ejemplo, el host `lustre_01` utiliza `host_vars/lustre_01.yml`, y `netapp_01` utiliza `host_vars/netapp_01.yml`. Incluye el grupo MGS solo en el primer bloque de construcción. Incluye los grupos MDS en el bloque de construcción base y en cada bloque de construcción de expansión MDT+OST; omite los grupos MDS únicamente de los bloques de construcción de expansión que sean solo OST. Registra los objetivos de expansión con respecto al MGS existente mediante `mgsnode`. |

Repite la configuración para cada host y cada bloque de construcción. Crea un `host_vars/lustre_NN.yml` archivo por cada servidor Lustre y un `host_vars/netapp_NN.yml` archivo por cada matriz, y repite los grupos de inventario para cada bloque de construcción adicional.

Las guías de implementación cargan las credenciales de array, Pacemaker y BMC desde el archivo compartido `building_blocks/passwords.yml`. Rellena este archivo y encríptalo con Ansible Vault antes de la implementación, o sobrescribe los valores de forma segura en tiempo de ejecución con `--extra-vars`. No guardes credenciales en texto plano en el control de versiones.

> Las direcciones IP, los nombres de host, los nombres de interfaz y el ajuste de tamaño de volúmenes en las plantillas de implementación son ejemplos. Modifica todos los archivos de inventario del entorno antes de ejecutar cualquier playbook.

**Después de que termines**

"Implementa el clúster Lustre de alta disponibilidad y los clientes".

## Implementa el clúster de alta disponibilidad de Lustre y los clientes

Prepara el sistema operativo del servidor, instala NVIDIA DOCA-OFED, implementa el clúster Lustre de alta disponibilidad y configura los clientes de Lustre usando los playbooks de Ansible de NetApp.

Antes de empezar, completa y asegura el inventario que se describe en "Personaliza el inventario de Ansible de Lustre".

### Implementa el clúster de alta disponibilidad de Lustre

**Pasos**

1. Desde el directorio `building_blocks/playbooks` de tu copia de trabajo, prepara el sistema operativo en los servidores Lustre. Sustituye `<inventory_path>` por la ruta de acceso a tu directorio de inventario personalizado:

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

2. Instala NVIDIA DOCA-OFED en cada servidor Lustre. Compila los módulos del kernel con el kernel que instalaste en el paso anterior. Sigue el procedimiento para tu kernel en "Instalación y actualización de NVIDIA DOCA-Host". El siguiente ejemplo muestra una instalación de DOCA-OFED en RHEL o Rocky Linux.

   a. Instala el paquete del repositorio principal de DOCA y actualiza la caché de paquetes. Sustituye `<doca_host_package>` por el paquete principal de DOCA correspondiente a tu sistema operativo y arquitectura:

      ```
      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 los módulos del kernel DOCA para el kernel que está en ejecución. El script muestra la ruta del paquete que crea:

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

   c. Instala el paquete del repositorio de módulos del kernel que creó el script y luego actualiza la caché de paquetes. Reemplaza `<doca_kernel_repo>` con la ruta del paso anterior:

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

   d. Instala los paquetes de DOCA-OFED para el espacio de usuario, el kernel y NVMe. Sustituye `<kernel_version>` por la versión del kernel que tengas en uso:

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

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

3. Ejecuta el playbook de implementación principal:

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

> Ajusta `--forks` según el tamaño de la implementación para controlar cuántos hosts configura Ansible en paralelo. Por ejemplo, añade `--forks 20` para un clúster más grande.

El manual de procedimientos aprovisiona grupos de volúmenes E-Series o pools y volúmenes DDP, configura la conectividad de host NVMe-oF, formatea y monta los targets Lustre, configura LNet, aplica ajustes de rendimiento y configura los recursos HA de Pacemaker.

> Una implementación con un bloque de construcción forma un clúster de dos nodos de Pacemaker. Al nodo A se le asigna un voto adicional en caso de conmutación por error. Para lograr el quórum en un clúster de dos nodos, considera configurar un qdevice. Cuando una implementación contiene varios bloques de construcción, cada bloque de construcción aporta un par de servidores de dos nodos al clúster de Pacemaker más grande, hasta el límite de clúster documentado. Revisa la configuración de fencing y quórum en el "Guía de usuario de la administración de HA" para que el clúster pueda recuperar de forma segura los objetivos durante una falla de nodo.

### Implementar clientes Lustre

Implementa clientes de Lustre en cualquier sistema que requiera acceso al sistema de archivos personalizando la plantilla de implementación del cliente y ejecutando el playbook del cliente.

#### Pasos

1. Selecciona un sistema operativo cliente y un kernel compatibles en "Matriz de compatibilidad de Lustre". Compila e instala los paquetes del cliente Lustre para ese kernel desde "netapp-lustre" si aún no están instalados.
2. Crea una copia de trabajo del directorio de plantillas `clients`, incluido su archivo compartido `passwords.yml`, y selecciona la plantilla que coincida con tu fabric:
   - **InfiniBand:** "IB_lustre_cluster_clients"
   - **RoCE:** "ROCE_lustre_cluster_clients"
3. Personaliza `lustre_client_inventory.yml`, `host_vars` y `group_vars` para tus clientes. Las referencias de las líneas de la tabla apuntan a los archivos de plantilla de clientes RoCE en la etiqueta `ansible-lustre release/1.0.0`; la plantilla de cliente InfiniBand usa las mismas variables salvo donde se indique lo contrario.

| Campo | Variable de Ansible | Archivo de plantilla (`release/1.0.0`) | Ámbito | Notas |
|---|---|---|---|---|
| Hosts cliente | `lustre_clients` | "lustre_client_inventory.yml #L4" | Por cliente | Enumera el nombre de host de cada cliente. |

| Campo | Variable de Ansible | Archivo de plantilla (`release/1.0.0`) | Ámbito | Notas |
|---|---|---|---|---|
| Usuario SSH del cliente y contraseña become | `ssh_client_user` / `ssh_client_become_pass` | "passwords.yml#L2" | A nivel de clúster | Establece la contraseña «become» solo si el usuario SSH no es `root`. |
| Interfaces LNet | `eseries_lustre_lnet` | "group_vars/all.yml#L7" | A nivel de clúster | Nombres de las interfaces LNet del lado del cliente (valores en L13). |
| Montar los NID de MGS | `lustre_client_mounts.mgsnode` | "group_vars/all.yml#L17" | A nivel de clúster | NIDs de MGS utilizados para montar el sistema de archivos (valores en L20). |
| Nombre del sistema de archivos | `lustre_client_mounts.fsname` | "group_vars/all.yml#L21" | A nivel de clúster | Debe coincidir con el clúster `fsname`. |
| Punto de montaje | `lustre_client_mounts.mount_point` | "group_vars/all.yml#L22" | A nivel de clúster | Directorio de montaje del cliente. |
| Dirección IP de gestión de cliente | `ansible_host` | "host_vars/client_01.yml#L4" | Repetir para cada cliente | Un archivo `host_vars/client_NN.yml` por cliente. |
| Interfaces de la estructura de red de almacenamiento | `eseries_roce_interfaces` | "host_vars/client_01.yml#L10" | Repetir para cada cliente | Direcciones de la interfaz front-end (valores en L15). La plantilla InfiniBand utiliza `eseries_ipoib_interfaces` aquí. |

Repite la configuración para cada cliente. Crea un `host_vars/client_NN.yml` archivo para cada cliente y añade el host correspondiente a `lustre_client_inventory.yml`. Rellena el archivo compartido `clients/passwords.yml` y encríptalo con Ansible Vault antes de ejecutar el playbook del cliente.

4. Ejecuta el playbook del cliente desde el directorio de plantillas del cliente:

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

La guía de configuración del cliente configura las interfaces de red del cliente y LNet, y puede crear una unidad de montaje persistente de systemd para el sistema de archivos Lustre.

**Después de que termines**

"Verifica la implementación".

## Verifica y amplía la implementación de Lustre

Comprueba el estado del clúster, los montajes de Lustre y la capacidad del sistema de archivos antes de su uso en producción, luego usa ese mismo inventario para añadir componentes básicos cuando lo necesites.

### Verifica la implementación

**Pasos**

1. Desde un servidor Lustre, comprueba que todos los recursos de Pacemaker se hayan iniciado en los nodos previstos:

```
pcs status
```

2. Desde un cliente de Lustre, comprueba que el sistema de archivos Lustre esté montado:

```
mount -t lustre
```

3. Desde un cliente de Lustre, comprueba la capacidad del sistema de archivos y que los destinos MDT y OST sean visibles:

```
lfs df -h
```

Para consultar la validación del rendimiento con IOR, mdtest y fio, consulta "Guía de tallas".

Para realizar pruebas de migración controlada de objetivos o de conmutación por error de HA, sigue los procedimientos en el "Guía de usuario de la administración de HA".

### Ampliar el sistema de archivos

Para aumentar la capacidad o el rendimiento de los metadatos, amplía tu inventario actual con un bloque de construcción adicional. No crees un inventario independiente para el nuevo bloque de construcción. El playbook asigna automáticamente índices MDT y OST únicos y registra los nuevos objetivos en el MGS existente.

**Pasos**

1. Añade los nuevos hosts, arrays y grupos de recursos (MDT+OST u OST únicamente) al mismo inventario y `host_vars` y `group_vars` archivos que usaste para la implementación original.
2. Vuelve a ejecutar el manual de implementación principal con el inventario actualizado.

Consulta "Arquitectura de la solución" para conocer las normas de escalado y "Guía de tallas" para obtener orientación sobre cuándo añadir cada tipo de bloque de construcción.

# Lustre con el almacenamiento NetApp E-Series: recursos relacionados

Utiliza estos enlaces para acceder a documentación, herramientas y software relacionados a la hora de planificar o ampliar una implementación de Lustre con almacenamiento E-Series de NetApp. Para conocer los pasos de dimensionamiento e implementación, consulta "Guía de tallas" y "Implementa la solución".

## NetApp recursos

- "Herramienta NetApp Interoperability Matrix Tool". Busca **E-Series Lustre**.
- "NetApp Ficha técnica de la cabina all-flash EF-Series"
- "NetApp documentación de SANtricity System Manager"
- "Colección ansible-lustre de NetApp". Para obtener orientación sobre el campo de inventario, consulta "Personaliza el inventario de Ansible de Lustre".
- "NetApp repositorio de código fuente de Lustre". NetApp ofrece paquetes RPM de servidor Lustre ya compilados para Rocky Linux 9.8 y Red Hat Enterprise Linux 9.8. Compila los paquetes de cliente de Lustre para el sistema operativo y el kernel del cliente a partir del código fuente de NetApp.

## Comunidad de Lustre

- "Sistema de archivos Lustre"
- "Matriz de compatibilidad de Lustre" para sistemas operativos y núcleos de cliente
- "Lustre Wiki: Ajuste de Lustre"

## Soluciones de infraestructura de IA relacionadas de NetApp

- "BeeGFS en NetApp con almacenamiento de la serie E"
- "Implementación de IBM Spectrum Scale con el almacenamiento E-Series de NetApp"
- "NVIDIA DGX SuperPOD con la serie EF de NetApp"

## Información de copyright



## Información de la marca comercial