採用E系列儲存設備的NetApp BeeGFS

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

目錄

採用E系列儲存設備的NetApp BeeGFS

開始使用

本網站所含內容

本網站說明如何使用NetApp認證架構（NVA）和自訂架構、在NetApp上部署及管理BeeGFS。NVA設計經過徹底測試、並提供客戶參考組態與規模調整指引、以將部署風險降至最低、並加速上市時間。NetApp也支援在NetApp硬體上執行自訂BeeGFS架構、讓客戶與合作夥伴能夠靈活設計檔案系統、以滿足各種需求。這兩種方法都運用Ansible進行部署、提供類似應用裝置的方法、在靈活的硬體範圍內、以任何規模管理BeeGFS。

詞彙與概念

下列術語與概念適用於NetApp上的BeeGFS解決方案。

如"管理BeeGFS叢集"需與 BeeGFS 高可用度（ HA ）叢集互動的專屬詞彙與概念、請參閱一節。

期限	說明
AI	人工智慧：
Ansible Control Node	用於執行 Ansible CLI 的實體或虛擬機器。
可Ansible Inventory	目錄結構包含Yaml檔案、可用來描述所需的BeeGFS HA叢集。
BMC	基礎板管理控制器。有時稱為服務處理器。
區塊節點	E 系列儲存系統。
用戶端	HPC叢集中執行需要使用檔案系統之應用程式的節點。有時也稱為運算或GPU節點。
DL	深度學習：
檔案節點	BeeGFS檔案伺服器。
HA	高可用度：
HIC	主機介面卡。
高效能運算	高效能運算：

期限	說明
HPC型工作負載	HPC型工作負載的特點通常是多個運算節點或GPU都需要同時存取相同的資料集、以利分散式運算或訓練工作。這些資料集通常是由大型檔案所組成、應跨越多個實體儲存節點進行等量分佈、以消除傳統硬體瓶頸、避免同時存取單一檔案。
ML	機器學習：
NLP	自然語言處理：
NLU	自然語言理解：
NVA	NetApp驗證架構（NVA）方案針對特定工作負載和使用案例、提供參考組態和規模調整指引。這些解決方案經過徹底測試、旨在將部署風險降至最低、並加速上市時間。
儲存網路/用戶端網路	用於用戶端與BeeGFS檔案系統通訊的網路。這通常是用於HPC叢集節點之間平行訊息傳遞介面（MPI）和其他應用程式通訊的相同網路。

使用通過驗證的架構

總覽與需求

解決方案總覽

BeeGFS on NetApp解決方案結合BeeGFS平行檔案系統與NetApp EF600儲存系統、打造可靠、可擴充且具成本效益的基礎架構、可跟上嚴苛工作負載的腳步。

NVA方案

NetApp上的BeeGFS解決方案是NetApp驗證架構（NVA）方案的一部分、可為客戶提供特定工作負載和使用案例的參考組態和規模調整指導。NVA解決方案經過徹底測試與設計、可將部署風險降至最低、並加速上市時間。

設計總覽

NetApp 上的 BeeGFS 解決方案是一種可擴充的建置區塊架構、可針對各種嚴苛的工作負載進行設定。無論是處理許多小型檔案、管理大量檔案作業、或是混合式工作負載、都能自訂檔案系統以滿足這些需求。高可用度的設計採用兩層硬體結構、可在多個硬體層進行自動容錯移轉、確保效能一致、即使在部分系統降級期間也是如此。BeeGFS 檔案系統可在不同的 Linux 套裝作業系統中提供高效能且可擴充的環境、並為用戶端提供單一易存取的儲存命名空間。如需詳細資訊，請參閱 "架構總覽"。

使用案例

下列使用案例適用於NetApp上的BeeGFS解決方案：

- NVIDIA DGX SuperPOD 系統採用 DGX ，搭配 A100 ， H100 ， H200 和 B200 GPU 。
- 人工智慧（AI）包括機器學習（ML）、深度學習（DL）、大規模自然語言處理（NLP）、以及自然語言理解（N5U）。如需詳細資訊、請參閱 "BeeGFS for AI：事實與虛構"。
- 高效能運算（HPC）、包括透過MPI（訊息傳遞介面）和其他分散式運算技術加速的應用程式。如需詳細資訊、請參閱 "為什麼BeeGFS超越HPC"。
- 應用程式工作負載的特徵為：
  - 讀取或寫入大於1GB的檔案
  - 由多個用戶端（10s、100s和1000s）讀取或寫入同一個檔案
- 多TB或數PB資料集。
- 需要單一儲存命名空間的環境、可針對大型與小型檔案的組合進行最佳化。

效益

在NetApp上使用BeeGFS的主要優點包括：

- 通過驗證的硬體設計可提供完整的硬體與軟體元件整合、確保可預測的效能與可靠性。
- 使用Ansible進行部署與管理、以達到簡化與大規模一致的目標。
- 使用E系列效能分析器和BeeGFS外掛程式提供監控和觀察能力。如需詳細資訊、請參閱 "介紹監控NetApp E系列解決方案的架構"。

- 高可用度採用共享磁碟架構、提供資料持久性與可用度。
- 使用Container和Kubernetes支援現代化的工作負載管理與協調。如需詳細資訊、請參閱 "Kubernetes與BeeGFS會面：這是一段符合未來需求的投資故事"。

架構總覽

NetApp上的BeeGFS解決方案包含架構設計考量、可用來判斷支援已驗證工作負載所需的特定設備、纜線和組態。

建置區塊架構

BeeGFS檔案系統可根據儲存需求以不同方式進行部署和擴充。例如、主要包含大量小型檔案的使用案例、將可從額外的中繼資料效能和容量中獲益、而較少大型檔案的使用案例、則可能會讓實際檔案內容的儲存容量和效能更高。這些多重考量因素會影響平行檔案系統部署的不同層面、進而增加設計和部署檔案系統的複雜度。

為了因應這些挑戰、NetApp設計了標準建置區塊架構、用於橫向擴充這些層面。通常、BeeGFS建置區塊會部署在三種組態設定檔中的其中一種：

- 單一基礎建置區塊、包括BeeGFS管理、中繼資料和儲存服務
- BeeGFS中繼資料加上儲存建置區塊
- BeeGFS僅儲存建置區塊

這三個選項之間唯一的硬體變更是使用較小的磁碟機來處理BeeGFS中繼資料。否則、所有組態變更都會透過軟體套用。使用Ansible做為部署引擎、為特定建置區塊設定所需的設定檔、可讓組態工作變得簡單明瞭。

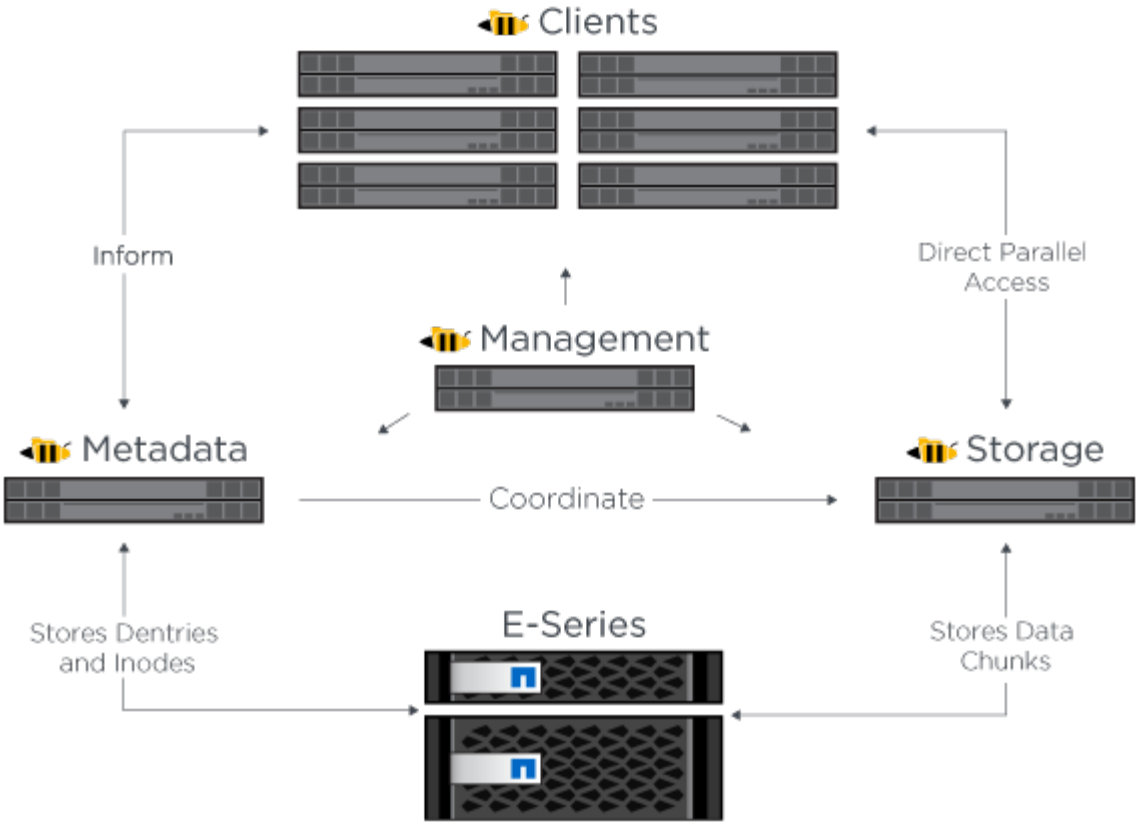
如需詳細資料、請參閱 [已驗證硬體設計]。

檔案系統服務

BeeGFS檔案系統包含下列主要服務：

- *管理服務。*註冊並監控所有其他服務。
- *儲存服務。*儲存稱為資料區塊檔案的分散式使用者檔案內容。
- *中繼資料服務。*會追蹤檔案系統配置、目錄、檔案屬性等。
- *用戶端服務。*掛載檔案系統以存取儲存的資料。

下圖顯示了與NetApp E系列系統搭配使用的BeeGFS解決方案元件和關係。



作為平行檔案系統、BeeGFS會將其檔案等量磁碟區化到多個伺服器節點上、以最大化讀寫效能和擴充性。伺服器節點可共同運作、提供單一檔案系統、讓其他伺服器節點（通常稱為_Clients_）同時掛載及存取。這些用戶端可以像NTFS、XFS或ext4等本機檔案系統一樣、查看及使用分散式檔案系統。

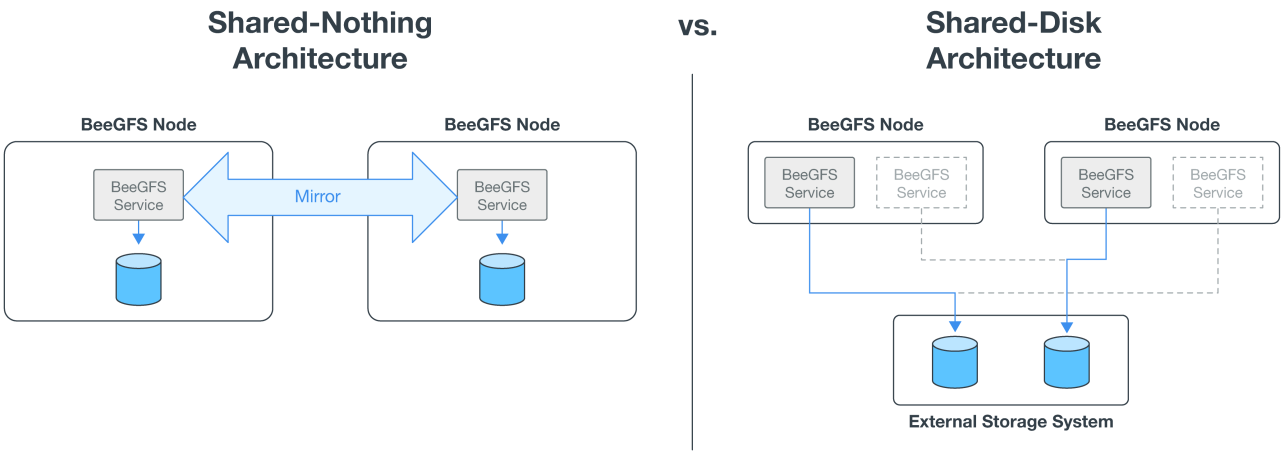
這四項主要服務可在多種支援的Linux套裝作業系統上執行、並可透過任何支援TCP/IP或RDMA的網路進行通訊、包括InfiniBand（IB）、OMNI-Path（opa）和RDMA over Converged Ethernet（roce）。BeeGFS伺服器服務（管理、儲存及中繼資料）是使用者空間精靈、而用戶端則是原生核心模組（無修補程式）。所有元件均可在不重新開機的情況下安裝或更新、而且您可以在同一個節點上執行任何服務組合。

HA架構

NetApp的BeeGFS透過NetApp硬體打造完全整合的解決方案、實現共享磁碟高可用度（HA）架構、擴充BeeGFS企業版的功能。

> 雖然BeeGFS社群版本可以免費使用、但企業版需要向NetApp等合作夥伴購買專業支援訂閱合約。企業版允許使用多項額外功能、包括恢復能力、配額強制和儲存資源池。

下圖比較了共享無共享和共享磁碟HA架構。



如需詳細資訊、請參閱 "發表NetApp支援的BeeGFS高可用度"。

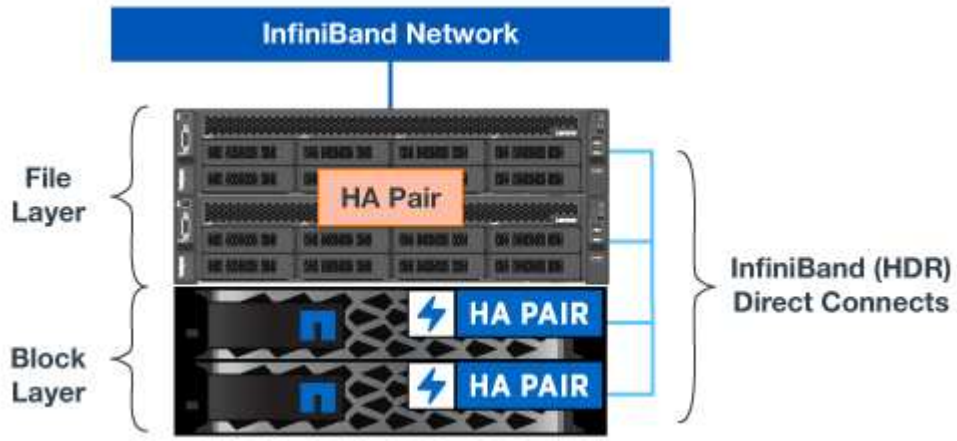
已驗證節點

NetApp 解決方案上的 BeeGFS 已驗證下列節點。

節點	硬體	詳細資料
區塊	NetApp EF600 儲存系統	專為嚴苛工作負載所設計的高效能全 NVMe 2U 儲存陣列。
檔案	Lenovo ThinkSystem SR665 V3 伺服器	雙插槽 2U 伺服器、採用 PCIe 5.0 、雙 AMD EPYC 9124 處理器。如需 Lenovo SR665 V3 的詳細資訊、請參閱 "聯想的網站"。
	Lenovo ThinkSystem SR665 伺服器	雙插槽 2U 伺服器、採用 PCIe 4.0 、雙 AMD EPYC 7003 處理器。如需 Lenovo SR665 的詳細資訊、請參閱 "聯想的網站"。

已驗證硬體設計

此解決方案的建置區塊（如下圖所示）使用通過驗證的檔案節點伺服器作為 BeeGFS 檔案層、並使用兩個 EF600 儲存系統做為區塊層。

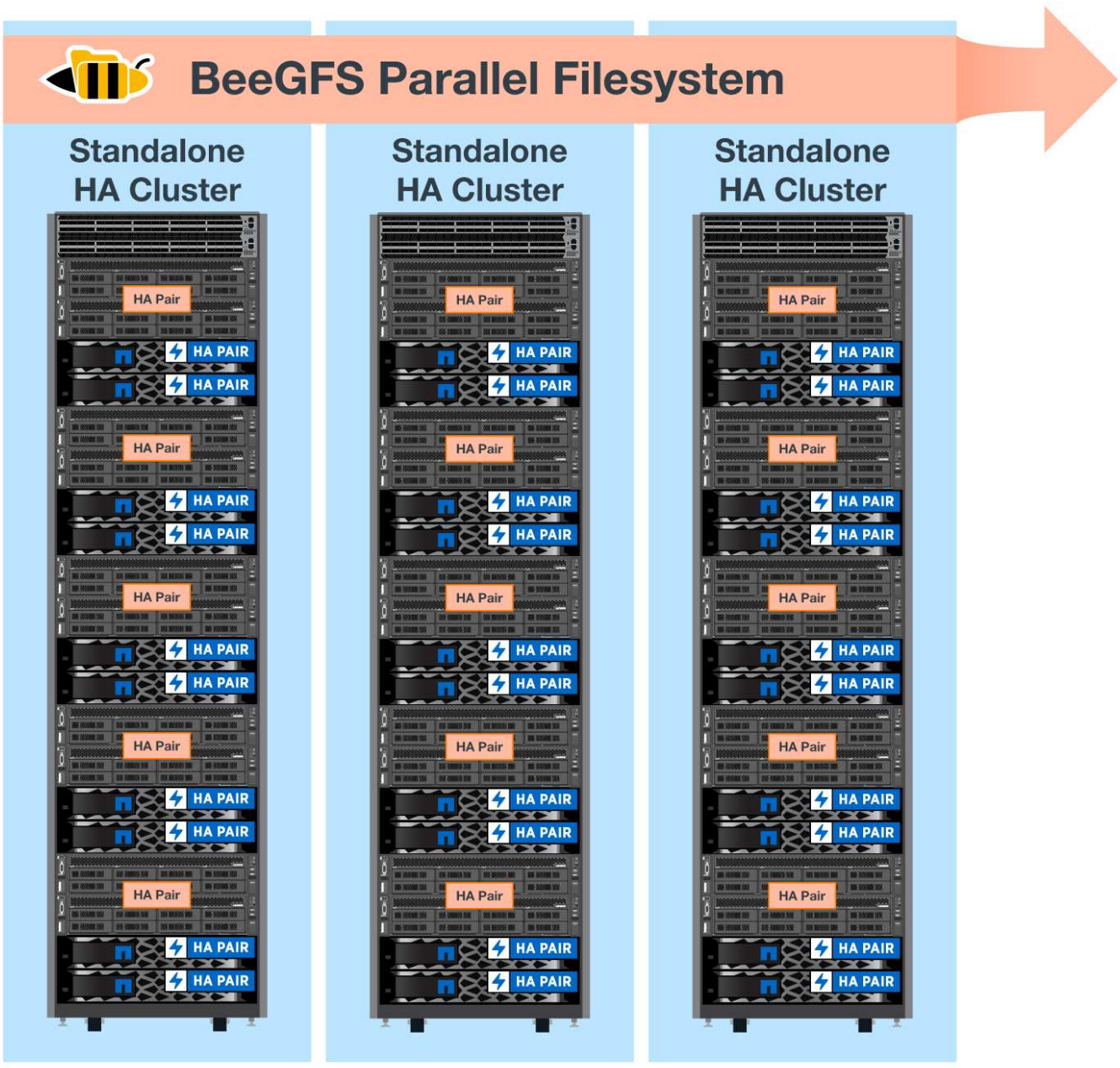


NetApp上的BeeGFS解決方案可在部署中的所有建置區塊上執行。部署的第一個建置區塊必須執行 BeeGFS 管理、中繼資料和儲存服務（稱為基礎建置區塊）。所有後續的建置區塊均可透過軟體進行設定、以擴充中繼資料

和儲存服務、或是僅提供儲存服務。這種模組化方法可根據工作負載的需求擴充檔案系統、同時使用相同的基礎硬體平台和建置區塊設計。

最多可部署五個建置區塊、形成獨立式 Linux HA 叢集。如此可利用 Pacemaker 最佳化資源管理、並與電暈器同步保持高效率同步。這些獨立式 BeeGFS HA 叢集中有一或多個是結合在一起的、可建立 BeeGFS 檔案系統、讓用戶端以單一儲存命名空間的形式存取。在硬體方面、單一 42U 機架最多可容納五個建置區塊、以及兩個用於儲存 / 資料網路的 1U InfiniBand 交換器。請參閱下圖以取得視覺呈現。

在容錯移轉叢集中建立仲裁所需的建置區塊至少有兩個。雙節點叢集具有可能會阻止容錯移轉成功的限制。您可以將第三個裝置整合為 tiebreaker 來設定雙節點叢集、但本文件並未說明該設計。



Ansible

NetApp上的BeeGFS是使用Ansible Automation（位於GitHub和Ansible Galaxis）（BeeGFS收藏可從取得）來

交付及部署 "Ansible Galaxy" 和 "NetApp的E系列GitHub"）。雖然Ansible主要是針對用來組裝BeeGFS建置區塊的硬體進行測試、但您可以設定它在任何使用支援Linux套裝作業系統的x86型伺服器上執行。

如需詳細資訊、請參閱 "部署BeeGFS搭配E系列儲存設備"。

技術需求

若要在 NetApp 上實作 BeeGFS 解決方案、請確保您的環境符合本文件所述的技術需求。

硬體需求

開始之前、請先確認您的硬體符合下列規格、以便在 NetApp 解決方案上進行 BeeGFS 的單一第二代建置區塊設計。特定部署的確切元件可能會因客戶需求而異。

數量	硬體元件	需求
2.	BeeGFS 檔案節點	每個檔案節點都應符合或超過建議檔案節點的規格、以達到預期的效能。 • 建議的檔案節點選項： * • * Lenovo ThinkSystem SR665 V* ◦ * 處理器： * 2 個 AMD EPYC 9124 16C 3.0 GHz （設定為兩個 NUMA 區域）。 ◦ * 記憶體： * 256GB （ 16x 16GB TruDDR5 4800MHz RDIMM ） ◦ * PCIe 擴充： * 四個 PCIe Gen5 x16 插槽（每個 NUMA 區域兩個） ◦ 雜項： ▪ 適用於作業系統的 RAID 1 中有兩個磁碟機（ 1TB 7.2K SATA 或更高） ▪ 1GbE 連接埠、用於頻內 OS 管理 ▪ 1GbE BMC 搭配 Redfish API 、用於頻外伺服器管理 ▪ 雙熱交換電源供應器和效能風扇
2.	E-Series 區塊節點（ EF600 陣列）	• 記憶體： * 256GB （每個控制器 128GB ）。* 介面卡： * 2 埠 200GB/HDR （ NVMe / IB ）。* 磁碟機： * 設定為符合所需的中繼資料和儲存容量。
8.	InfiniBand 主機卡介面卡（適用於檔案節點）。	主機卡適配器可能會根據檔案節點的伺服器型號而有所不同。驗證檔案節點的建議包括： • * Lenovo ThinkSystem SR665 V3 伺服器： * ◦ MCX755106AS-Heat ConnectX-7 、 NDR200 、 QSFP112 、 2 埠、 PCIe Gen5 x16 、 InfiniBand 介面卡

數量	硬體元件	需求
1.	儲存網路交換器	儲存網路交換器的 InfiniBand 速度必須達到 200Gb/s 。建議的交換器機型包括： • * NVIDIA QM9700 Quantum 2 NDR InfiniBand 交換器 * • * NVIDIA MQM8700 Quantum HDR InfiniBand 交換器 *

纜線需求

- 從區塊節點直接連線至檔案節點。 *

數量	產品編號	長度
8.	MCP1650-H001E30 （ NVIDIA 被動銅線、 QSFP56 、 200Gb/s ）	1M

- 從檔案節點到儲存網路交換器的連線。 *根據您的 InfiniBand 儲存交換器、從下表中選取適當的纜線選項。+ 建議的纜線長度為 2 公尺；不過、這可能會因客戶的環境而異。

交換器模式	纜線類型	數量	產品編號
NVIDIA QM9700	主動式光纖（包括收發器）	2.	MMA4Z00-NS （多重模式， IB/ETH ， 800Gb/s 2x400Gb/s 雙埠 OFP ）
		4	MFP7E20-Nxxx （多重模式， 4 通道對兩條 2 通道分離器光纖纜線）
		8.	MMA1Z00-NS400 （多重模式， IB/ETH ， 400GB / 秒單埠 QSFP-112 ）
	被動銅	2.	MCP7Y40-N002 （ NVIDIA 被動銅線分離器纜線， InfiniBand 800Gb/s 至 4x 200Gb/s ， OSFP 至 4x QSFP112 ）
NVIDIA MQM8700	主動式光纖	8.	MFS1S00-H003E （ NVIDIA 主動式光纖纜線、 InfiniBand 200Gb/s 、 QSFP56 ）
	被動銅	8.	MCP1650-H002E26 （ NVIDIA 被動式銅線、 InfiniBand 200Gb/s 、 QSFP56 ）

軟體與韌體需求

為了確保可預測的效能和可靠性， NetApp 解決方案上的 BeeGFS 版本會使用特定版本的軟體和韌體元件進行測試。實作解決方案需要這些版本。

檔案節點需求

軟體	版本
Red Hat Enterprise Linux（RHEL）	RHEL 9.4 高可用性實體伺服器（雙插槽）。*附註：*檔案節點需要有效的 Red Hat Enterprise Linux Server 訂閱和 Red Hat Enterprise Linux 高可用性附加元件。
Linux核心	5.14.0-427.42.1.el9_4.x86_64

軟體	版本
HCA 韌體	**ConnectX-7 HCA** 韌體 FW：28.45.1200 + PXE：3.7.0500 + UEFI：14.38.0016 • ConnectX-6 HCA 韌體 * 韌體： 20.43.2566 + PXE ： 3.7.0500 + UEFI ： 14.37.0013

EF600區塊節點需求

軟體	版本
作業系統SANtricity	11.90R3
NVSRAM	N6000-890834-D02.dp
磁碟機韌體	最新版本適用於使用中的磁碟機機型。請參閱"E-Series 磁碟韌體站台"。

軟體部署需求

下表列出在以Ansible為基礎的BeeGFS部署中、自動部署的軟體需求。

軟體	版本
BeeGFS	7.4.6
電暈器同步	3.1.8-1
起搏器	2.1.7-5.2
件	0.11.7-2
圍欄代理（紅魚 / APC ）	4.10.0-62
InfiniBand / RDMA驅動程式	MLNX_OFED_LINUX-23.10-3.2.2.1-LTS

Ansible控制節點需求

NetApp上的BeeGFS解決方案是從可存取的控制節點進行部署和管理。如需詳細資訊、請參閱 "Ansible文件"。

下表所列的軟體需求、是下列NetApp BeeGFS Ansible系列產品的特定版本。

軟體	版本
Ansible	10.x
Ansible 核心	>= 2.13.0
Python	3.10
其他Python套件	密碼編譯 -43.0.0 、 netaddr-1.3.0 、 ipaddr-2.2.0
NetApp E-Series BeeGFS Ansible Collection	3.2.0

檢視解決方案設計

設計總覽

需要特定設備、纜線和組態來支援BeeGFS on NetApp解決方案、此解決方案將BeeGFS平行檔案系統與NetApp EF600儲存系統結合在一起。

深入瞭解：

- "硬體組態"
- "軟體組態"
- "設計驗證"
- "規模調整準則"
- "效能調校"

衍生架構的設計與效能差異：

- "高容量建置區塊"

硬體組態

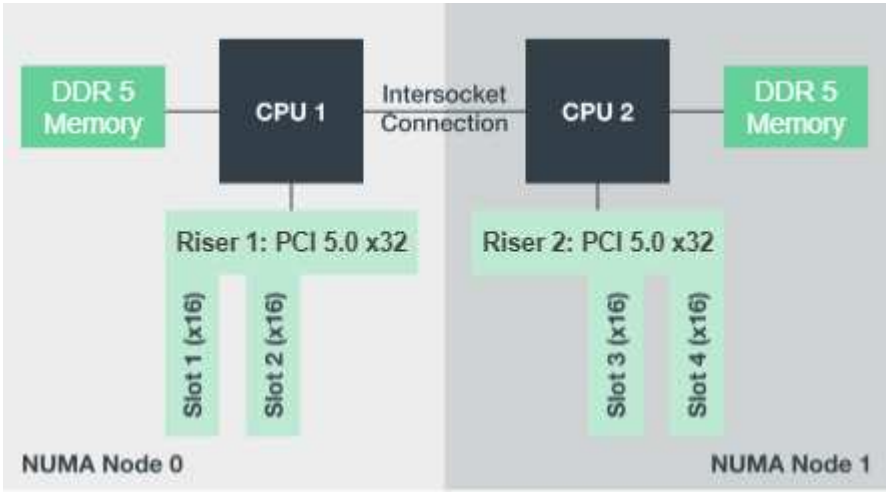
NetApp上BeeGFS的硬體組態包括檔案節點和網路纜線。

檔案節點組態

檔案節點有兩個CPU插槽、設定為獨立的NUMA區域、包括本機存取相同數量的PCIe插槽和記憶體。

InfiniBand介面卡必須安裝在適當的PCI擴充卡或插槽中、因此工作負載必須在可用的PCIe線道和記憶體通道之間取得平衡。您可以將個別BeeGFS服務的工作完全隔離到特定NUMA節點、藉此平衡工作負載。目標是從每個檔案節點取得類似的效能、就像是兩個獨立的單一插槽伺服器一樣。

下圖顯示檔案節點NUMA組態。



BeeGFS程序會固定在特定的NUMA區域、以確保所使用的介面位於相同的區域。此組態可避免透過插槽間連線

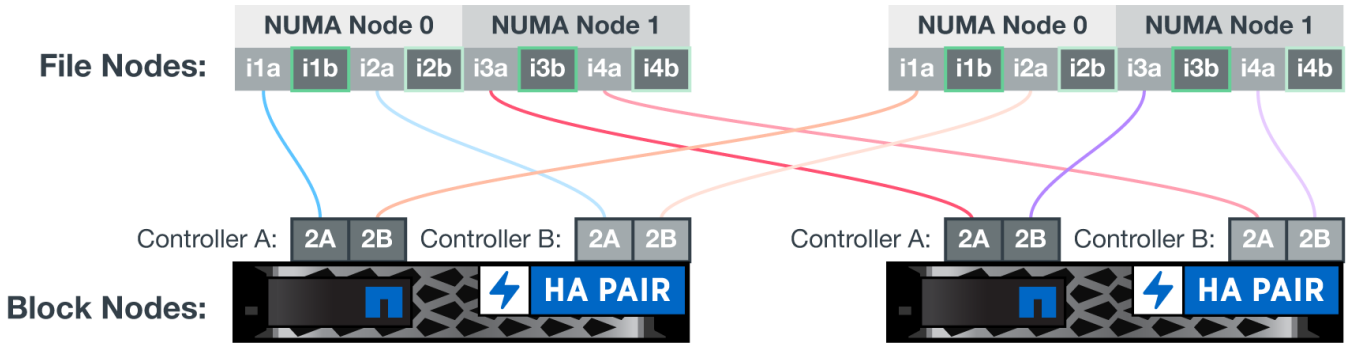
進行遠端存取。插槽之間的連線有時稱為QPI或GMI2連結、即使是在現代化的處理器架構中、也可能是使用高速度網路（例如HDRInfiniBand）時的瓶頸。

網路纜線組態

在建置區塊中、每個檔案節點都會使用總共四個備援InfiniBand連線、連接至兩個區塊節點。此外、每個檔案節點都有四個與InfiniBand儲存網路的備援連線。

在下圖中、請注意：

- 所有以綠色顯示的檔案節點連接埠均用於連接至儲存架構；所有其他的檔案節點連接埠則是直接連接至區塊節點。
- 特定NUMA區域中的兩個InfiniBand連接埠會連接到同一個區塊節點的A和B控制器。
- NUMA節點0中的連接埠一律連線至第一個區塊節點。
- NUMA節點1中的連接埠會連線至第二個區塊節點。



當使用分離器纜線將儲存交換器連接至檔案節點時、一條纜線應分出並連接至淡綠色的連接埠。另一條纜線應分出並連接至暗綠色的連接埠。此外、對於具有備援交換器的儲存網路、淡綠色的連接埠應連接至一台交換器、而深綠色的連接埠則應連接至另一台交換器。

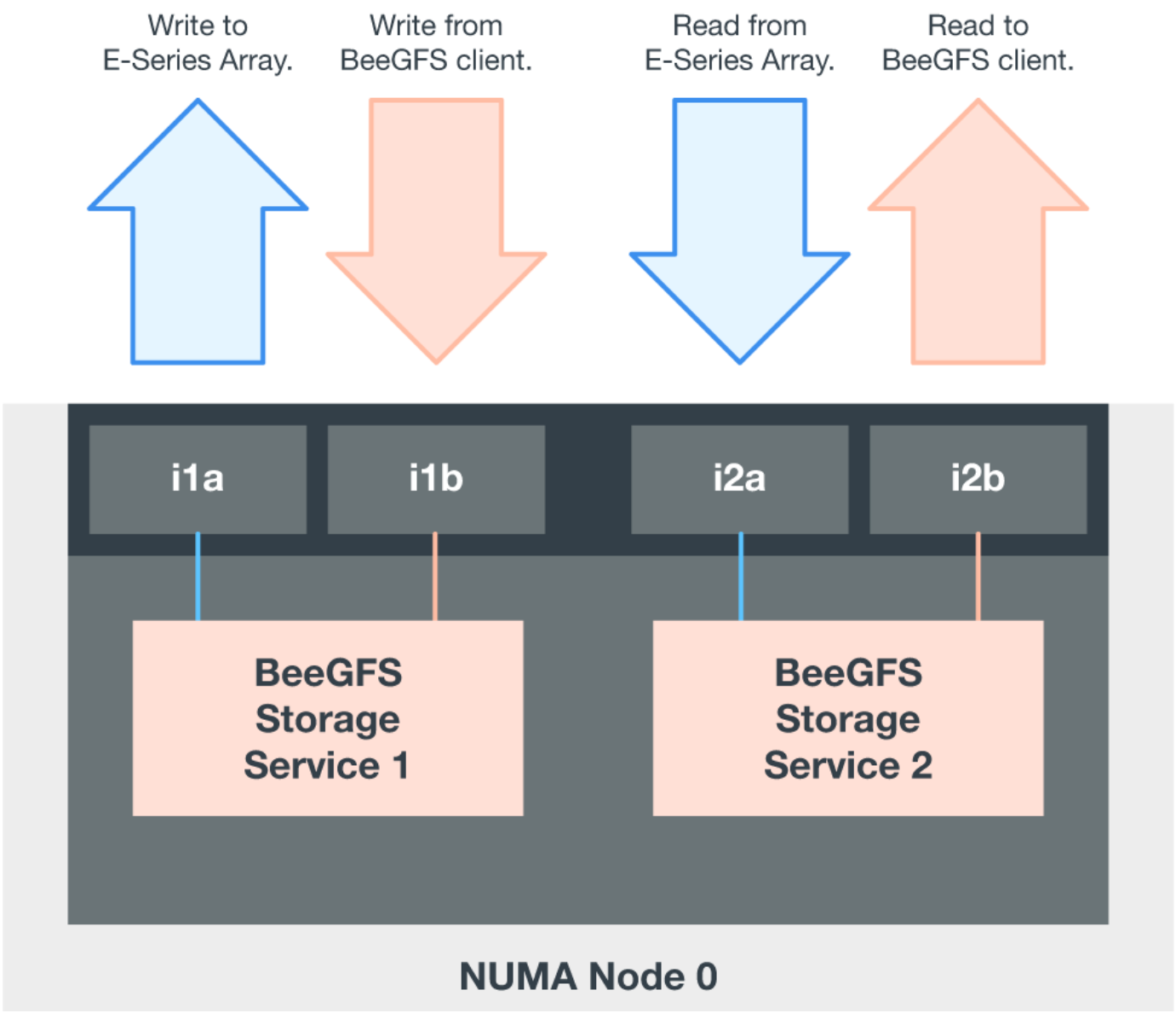
圖中所示的佈線組態可讓每個BeeGFS服務：

- 無論執行BeeGFS服務的檔案節點為何、都可在相同的NUMA區域中執行。
- 無論故障發生在何處、都要有次要的最佳路徑可通往前端儲存網路和後端區塊節點。
- 如果區塊節點中的檔案節點或控制器需要維護、請將效能影響降至最低。

利用頻寬的纜線

若要充分運用PCIe雙向頻寬、請確定每個InfiniBand介面卡上的一個連接埠連接至儲存架構、另一個連接埠則連接至區塊節點。

下圖顯示用於充分運用PCIe雙向頻寬的纜線設計。



對於每個BeeGFS服務、請使用相同的介面卡、將用戶端流量所使用的慣用連接埠、與服務磁碟區的主要擁有者區塊節點控制器路徑連線。如需詳細資訊、請參閱 "軟體組態"。

軟體組態

NetApp上BeeGFS的軟體組態包括BeeGFS網路元件、EF600區塊節點、BeeGFS檔案節點、資源群組和BeeGFS服務。

BeeGFS網路組態

BeeGFS網路組態包含下列元件。

- *浮動IP*浮動IP是一種虛擬IP位址、可動態路由傳送至同一個網路中的任何伺服器。多部伺服器可以擁有相同的浮動IP位址、但在任何指定時間、只能在一部伺服器上啟用。

  每個BeeGFS伺服器服務都有自己的IP位址、可視BeeGFS伺服器服務的執行位置而在檔案節點之間移動。此浮動IP組態可讓每個服務獨立容錯移轉至其他檔案節點。用戶端只需知道特定BeeGFS服務的IP位址、就

不需要知道目前執行該服務的檔案節點。

- * BeeGFS伺服器多重主頁組態*為了提高解決方案的密度、每個檔案節點都有多個儲存介面、其中IP設定在同一個IP子網路中。

  需要額外的組態、以確保此組態能與Linux網路堆疊正常運作、因為在預設情況下、如果某個介面的IP位在同一子網路中、則可在不同的介面上回應對該介面的要求。除了其他缺點、這種預設行為也使得無法正確建立或維護RDMA連線。

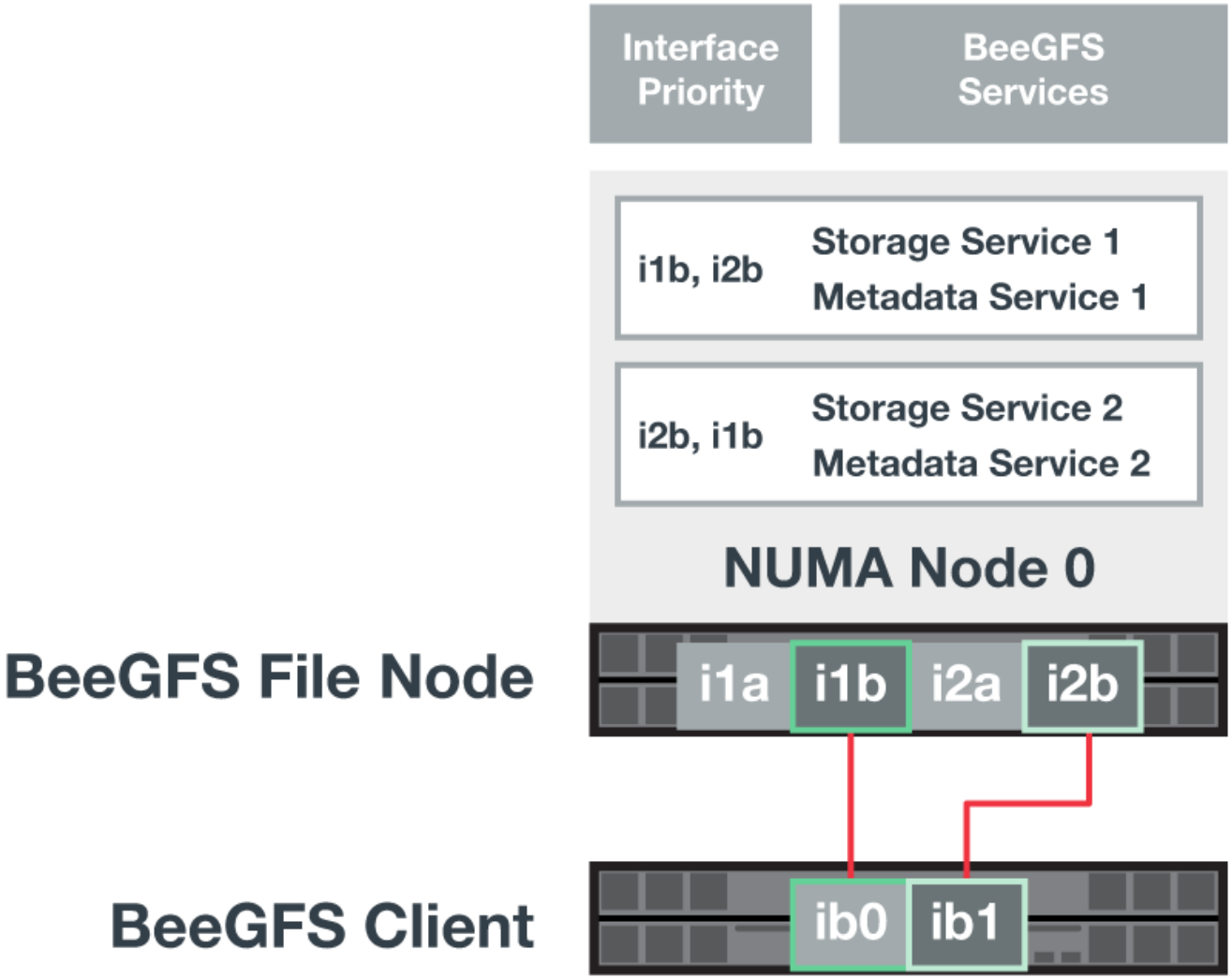
  Ansible型部署可處理反向路徑（RP）和位址解析傳輸協定（Arp）行為的強化、同時確保啟動和停止浮動IP；動態建立對應的IP路由和規則、讓多重主目錄網路組態正常運作。

- BeeGFS 用戶端多軌組態 * _Multi-rail 是指應用程式使用多個不同網路連線（或「 rail 」）來提高效能的能力。

  BeeGFS 實作多軌支援、可在單一 IPoIB 子網路中使用多個 IB 介面。此功能可在 RDMA NIC 之間啟用動態負載平衡等功能、以最佳化網路資源的使用。它也與 NVIDIA GPUDirect 儲存設備（ GDS ）整合、可提供更高的系統頻寬、並減少用戶端 CPU 的延遲和使用率。

  本文件提供單一 IPoIB 子網路組態的說明。支援雙 IPoIB 子網路組態、但並未提供與單一子網路組態相同的優勢。

  下圖顯示多個BeeGFS用戶端介面之間的流量平衡。



由於BeeGFS中的每個檔案通常會跨越多個儲存服務進行等量分佈、因此多重軌道組態可讓用戶端達到比單一InfiniBand連接埠更高的處理量。例如、下列程式碼範例顯示通用的檔案分段組態、可讓用戶端在兩個介面之間平衡流量：

+

```
root@beegfs01:/mnt/beegfs# beegfs-ctl --getentryinfo myfile
Entry type: file
EntryID: 11D-624759A9-65
Metadata node: meta_01_tgt_0101 [ID: 101]
Stripe pattern details:
+ Type: RAID0
+ Chunksize: 1M
+ Number of storage targets: desired: 4; actual: 4
+ Storage targets:
  + 101 @ stor_01_tgt_0101 [ID: 101]
  + 102 @ stor_01_tgt_0101 [ID: 101]
  + 201 @ stor_02_tgt_0201 [ID: 201]
  + 202 @ stor_02_tgt_0201 [ID: 201]
```

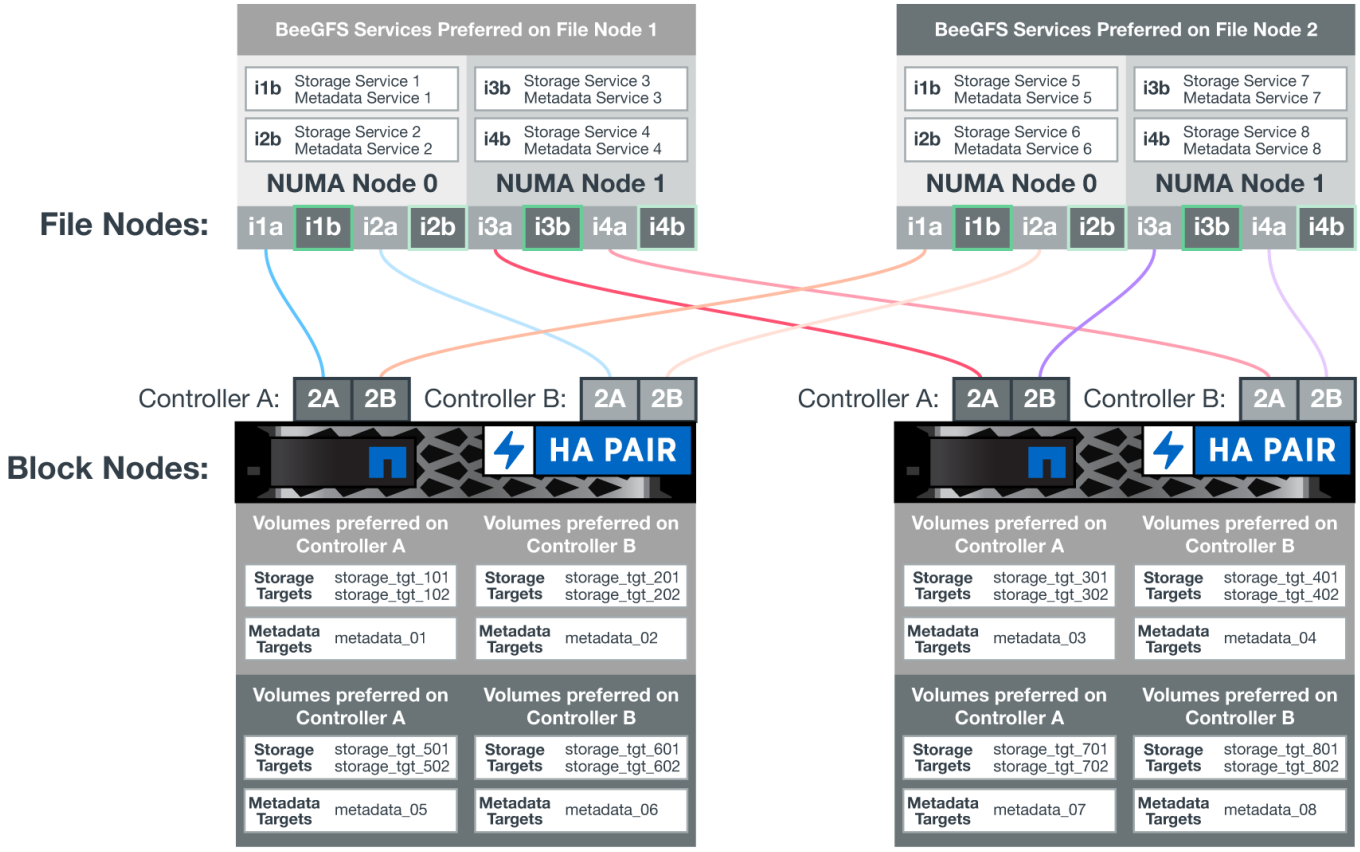
EF600區塊節點組態

區塊節點由兩個主動/主動式RAID控制器組成、可共用存取同一組磁碟機。一般而言、每個控制器擁有系統上設定的一半磁碟區、但可視需要接管其他控制器。

檔案節點上的多重路徑軟體可決定每個磁碟區的作用中最佳化路徑、並在纜線、介面卡或控制器故障時自動移至替代路徑。

下圖顯示EF600區塊節點中的控制器配置。

為了簡化共享磁碟HA解決方案、磁碟區會對應至兩個檔案節點、以便視需要彼此接管。下圖顯示如何設定BeeGFS服務和慣用磁碟區擁有權以達到最大效能的範例。每個BeeGFS服務左側的介面會指出用戶端和其他服務用來與其聯絡的偏好介面。



在前一個範例中、用戶端和伺服器服務偏好使用介面i1b與儲存服務1通訊。儲存服務1使用介面i1a做為首選路徑、以便在第一個區塊節點的控制器A上與其磁碟區（儲存設備_tgt_101、102）進行通訊。此組態可利用InfiniBand介面卡可用的全雙向PCIe頻寬、並從雙埠的HDRInfiniBand介面卡獲得比PCIe 4.0更好的效能。

BeeGFS檔案節點組態

BeeGFS檔案節點已設定為高可用度（HA）叢集、以便在多個檔案節點之間進行BeeGFS服務的容錯移轉。

HA叢集設計是以兩個廣泛使用的Linux HA專案為基礎：叢集成員資格的電暈器同步、以及叢集資源管理的起搏器。如需更多資訊、請參閱 "適用於高可用度附加元件的Red Hat訓練"。

NetApp撰寫並擴充數個開放式叢集架構（OCF）資源代理程式、讓叢集能夠智慧地啟動及監控BeeGFS資源。

BeeGFS HA叢集

一般而言、當您啟動BeeGFS服務（無論是否有HA）時、必須有幾個資源：

- 可連線服務的IP位址、通常由Network Manager設定。
- 作為BeeGFS儲存資料目標的基礎檔案系統。

  這些通常是在/etc/stab'中定義的、並由systemd掛載。

- 負責在其他資源準備就緒時啟動BeeGFS的系統服務。

  如果沒有其他軟體、這些資源只會在單一檔案節點上啟動。因此、如果檔案節點離線、則無法存取BeeGFS檔案系統的一部分。

由於多個節點可以啟動每個BeeGFS服務、因此心臟起搏器必須確保每個服務和相依資源一次只能在一個節點上執行。例如、如果兩個節點嘗試啟動相同的BeeGFS服務、則如果兩個節點都嘗試寫入基礎目標上的相同檔案、就會有資料毀損的風險。為了避免這種情況、心臟起搏器必須仰賴電暈器同步、才能在所有節點之間可靠地保持整體叢集的狀態同步、並建立仲裁。

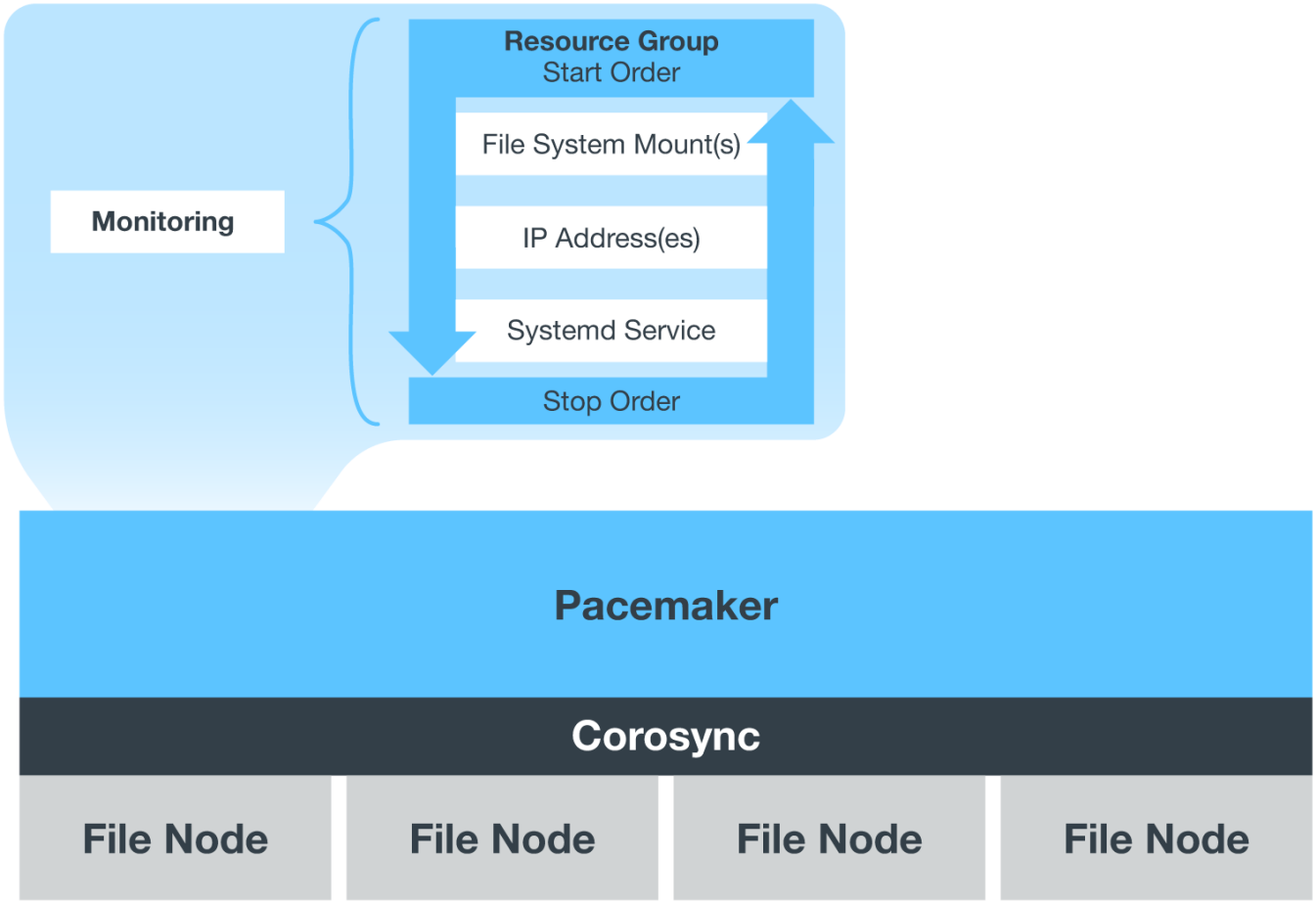
如果叢集發生故障、心臟起搏器會在另一個節點上反應並重新啟動BeeGFS資源。在某些情況下、心臟起搏器可能無法與原始故障節點通訊、以確認資源已停止。若要在重新啟動BeeGFS資源之前驗證節點是否已關閉、請先移除電源、使心臟起搏器從故障節點上關閉。

許多開放原始碼的屏障代理程式可讓心臟起搏器使用電力分配單元（PDU）或伺服器基板管理控制器（BMC）搭配API（例如RedfISH）來隔離節點。

當BeeGFS在HA叢集中執行時、所有BeeGFS服務和基礎資源都是由資源群組中的心臟起搏器管理。每個BeeGFS服務及其所依賴的資源都會設定成資源群組、以確保資源以正確的順序啟動和停止、並配置在同一個節點上。

對於每個BeeGFS資源群組、心臟起搏器都會執行自訂BeeGFS監控資源、負責偵測故障情況、並在特定節點上無法存取BeeGFS服務時、以智慧方式觸發容錯移轉。

下圖顯示由心臟起搏器控制的BeeGFS服務和相依性。



為了在同一個節點上啟動多個相同類型的BeeGFS服務、心臟起搏器已設定為使用多重模式組態方法來啟動BeeGFS服務。如需詳細資訊、請參閱 "多重模式的BeeGFS文件"。

由於BeeGFS服務必須能夠在多個節點上啟動、因此每項服務的組態檔（通常位於「/etc/beegfs」）會儲存在其中一個E系列磁碟區上、作為該服務的BeeGFS目標。如此一來、可能需要執行服務的所有節點都能存取特定BeeGFS服務的組態和資料。

```
# tree stor_01_tgt_0101/ -L 2
stor_01_tgt_0101/
├── data
│   ├── benchmark
│   ├── buddymir
│   ├── chunks
│   ├── format.conf
│   ├── lock.pid
│   ├── nodeID
│   ├── nodeNumID
│   ├── originalNodeID
│   ├── targetID
│   └── targetNumID
└── storage_config
    ├── beegfs-storage.conf
    ├── connInterfacesFile.conf
    └── connNetFilterFile.conf
```

設計驗證

NetApp解決方案BeeGFS的第二代設計已使用三種建置區塊組態設定檔進行驗證。

組態設定檔包括下列項目：

- 單一基礎建置區塊、包括BeeGFS管理、中繼資料和儲存服務。
- BeeGFS中繼資料加上儲存建置區塊。
- BeeGFS純儲存建置區塊。

建置區塊連接至兩台 NVIDIA Quantum InfiniBand （ MQM8700 ） 交換器。十個BeeGFS用戶端也連接到InfiniBand交換器、用來執行綜合基準測試公用程式。

下圖顯示用於驗證NetApp解決方案BeeGFS的BeeGFS組態。

BeeGFS檔案分段

平行檔案系統的一項優點是能夠跨越多個儲存目標、將個別檔案等量磁碟區、這可能代表相同或不同基礎儲存系統上的磁碟區。

在BeeGFS中、您可以根據每個目錄和每個檔案來設定分段、以控制用於每個檔案的目標數量、並控制用於每個檔案分段的chunksize（或區塊大小）。此組態可讓檔案系統支援不同類型的工作負載和I/O設定檔、而不需要重新設定或重新啟動服務。您可以使用「beegfs-CTL」命令列工具或使用分段API的應用程式來套用等量磁碟區設定。如需詳細資訊、請參閱的BeeGFS文件 "分段" 和 "分段API"。

為了達到最佳效能、在整個測試過程中都會調整等量磁碟區模式、並記錄每項測試所使用的參數。

IOR頻寬測試：多個用戶端

IOR頻寬測試使用OpenMPI來執行綜合I/O產生器工具IOR的平行工作（可從以下網站取得 "HPC GitHub"）跨所有10個用戶端節點、移至一或多個BeeGFS建置區塊。除非另有說明：

- 所有測試均使用直接I/O、傳輸大小為1MiB。
- BeeGFS檔案分段設定為1MB chunksize、每個檔案一個目標。

下列參數用於IOR、區段數經過調整、可將一個建置區塊的Aggregate檔案大小維持在5TiB、三個建置區塊的區段數維持在40TiB。

```
mpirun --allow-run-as-root --mca btl tcp -np 48 -map-by node -hostfile
10xnodes ior -b 1024k --posix.odirect -e -t 1024k -s 54613 -z -C -F -E -k
```

一個**BeeGFS**基礎（管理、中繼資料和儲存）建置區塊

下圖顯示單一BeeGFS基礎（管理、中繼資料和儲存）建置區塊的IOR測試結果。

BeeGFS中繼資料**+**儲存建置區塊

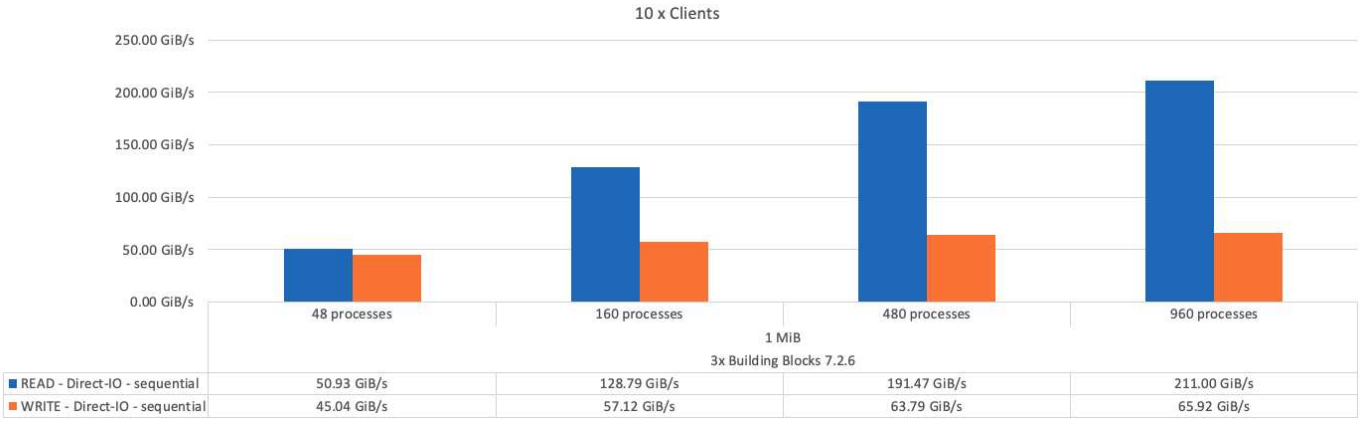
下圖顯示單一BeeGFS中繼資料+儲存建置區塊的IOR測試結果。

BeeGFS純儲存建置區塊

下圖顯示單一BeeGFS純儲存建置區塊的IOR測試結果。

三個**BeeGFS**建置區塊

下圖顯示使用三個BeeGFS建置區塊的IOR測試結果。



如預期、基礎建置區塊與後續中繼資料+儲存建置區塊之間的效能差異可忽略不計。比較中繼資料+儲存建置區塊與純儲存建置區塊、可看出讀取效能略有提升、因為使用額外的磁碟機做為儲存目標。不過、寫入效能並無顯著差異。若要達到更高的效能、您可以將多個建置區塊一起新增、以線性方式擴充效能。

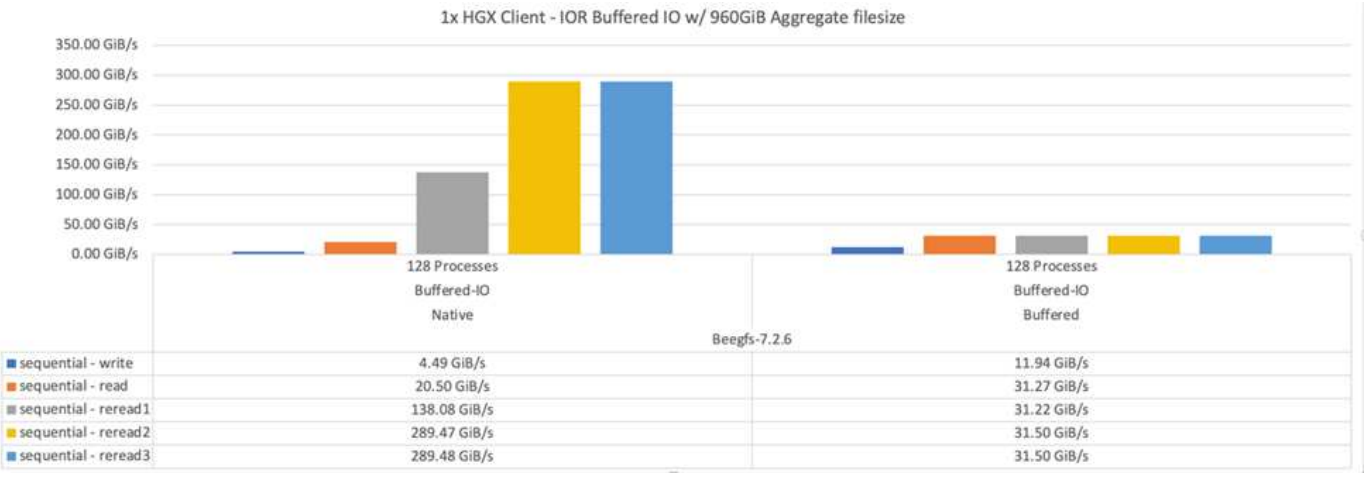
IOR頻寬測試：單一用戶端

IOR頻寬測試使用OpenMPI、使用單一高效能GPU伺服器執行多個IOR程序、以探索單一用戶端所能達到的效能。

此測試也會比較BeeGFS在用戶端設定為使用Linux核心分頁快取（「tuneFileCacheType = Native」）時的重新讀取行為和效能、以及預設的「緩衝」設定。

原生快取模式會使用用戶端上的Linux核心分頁快取、讓重新讀取作業從本機記憶體產生、而非透過網路重新傳輸。

下圖顯示使用三個BeeGFS建置區塊和單一用戶端的IOR測試結果。



這些測試的BeeGFS分段設定為1MB chunksize、每個檔案有八個目標。

雖然使用預設的緩衝模式時、寫入和初始讀取效能較高、但對於重讀相同資料多次的工作負載、原生快取模式可大幅提升效能。這項改善的重新讀取效能對於深度學習等工作負載來說非常重要、因為深度學習會在許多時期重讀相同的資料集多次。

中繼資料效能測試

中繼資料效能測試使用MDTest工具（包含在IOR中）來測量BeeGFS的中繼資料效能。測試使用OpenMPI在所有十個用戶端節點上執行平行工作。

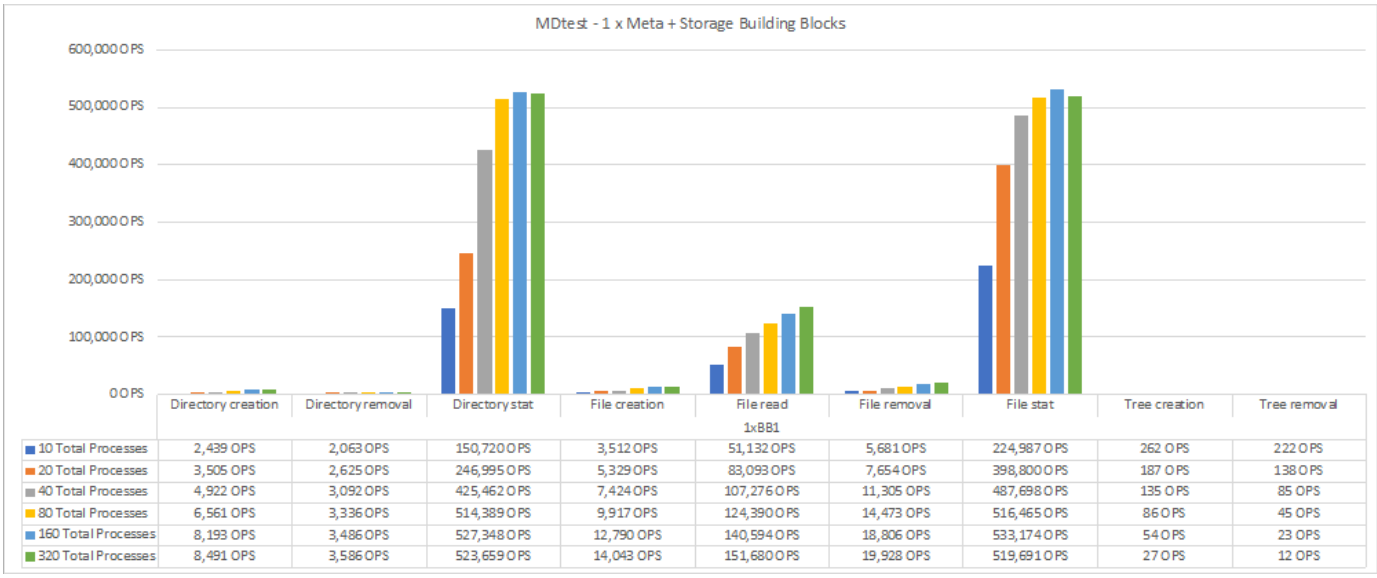
下列參數用於執行基準測試、其處理程序總數從10個增加到320個、步驟2個、檔案大小為4K。

```
mpirun -h 10xnodes –map-by node np $processes mdtest -e 4k -w 4k -i 3 -I
16 -z 3 -b 8 -u
```

中繼資料效能是先以一到兩個中繼資料+儲存建置區塊來測量、藉由新增額外的建置區塊來顯示效能如何擴充。

一個**BeeGFS**中繼資料**+**儲存建置區塊

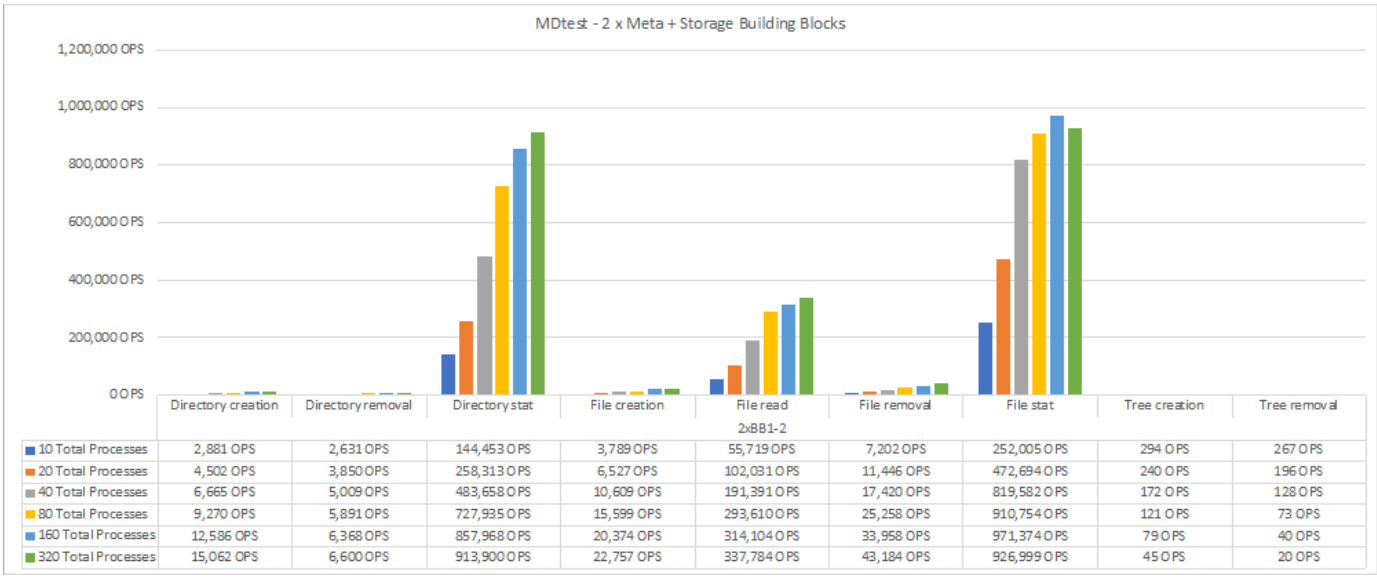
下圖顯示含有一個BeeGFS中繼資料+儲存建置區塊的MDTest結果。



	Directory creation	Directory removal	Directory stat	File creation	File read	File removal	File stat	Tree creation	Tree removal
					1xBB1				
10 Total Processes	2,439 OPS	2,063 OPS	150,720 OPS	3,512 OPS	51,132 OPS	5,681 OPS	224,987 OPS	262 OPS	222 OPS
20 Total Processes	3,505 OPS	2,625 OPS	246,995 OPS	5,329 OPS	83,093 OPS	7,654 OPS	398,800 OPS	187 OPS	138 OPS
40 Total Processes	4,922 OPS	3,092 OPS	425,462 OPS	7,424 OPS	107,276 OPS	11,305 OPS	487,698 OPS	135 OPS	85 OPS
80 Total Processes	6,561 OPS	3,336 OPS	514,389 OPS	9,917 OPS	124,390 OPS	14,473 OPS	516,465 OPS	86 OPS	45 OPS
160 Total Processes	8,193 OPS	3,486 OPS	527,348 OPS	12,790 OPS	140,594 OPS	18,806 OPS	533,174 OPS	54 OPS	23 OPS
320 Total Processes	8,491 OPS	3,586 OPS	523,659 OPS	14,043 OPS	151,680 OPS	19,928 OPS	519,691 OPS	27 OPS	12 OPS

兩個**BeeGFS**中繼資料**+**儲存建置區塊

下圖顯示含有兩個BeeGFS中繼資料+儲存建置區塊的MDTest結果。



	Directory creation	Directory removal	Directory stat	File creation	File read	File removal	File stat	Tree creation	Tree removal
					2xBB1-2				
10 Total Processes	2,881 OPS	2,631 OPS	144,453 OPS	3,789 OPS	55,719 OPS	7,202 OPS	252,005 OPS	294 OPS	267 OPS
20 Total Processes	4,502 OPS	3,850 OPS	258,313 OPS	6,527 OPS	102,031 OPS	11,446 OPS	472,694 OPS	240 OPS	196 OPS
40 Total Processes	6,665 OPS	5,009 OPS	483,658 OPS	10,609 OPS	191,391 OPS	17,420 OPS	819,582 OPS	172 OPS	128 OPS
80 Total Processes	9,270 OPS	5,891 OPS	727,935 OPS	15,599 OPS	293,610 OPS	25,258 OPS	910,754 OPS	121 OPS	73 OPS
160 Total Processes	12,586 OPS	6,368 OPS	857,968 OPS	20,374 OPS	314,104 OPS	33,958 OPS	971,374 OPS	79 OPS	40 OPS
320 Total Processes	15,062 OPS	6,600 OPS	913,900 OPS	22,757 OPS	337,784 OPS	43,184 OPS	926,999 OPS	45 OPS	20 OPS

功能驗證

在驗證此架構時、NetApp執行了數項功能測試、包括：

- 停用交換器連接埠、使單一用戶端InfiniBand連接埠故障。
- 停用交換器連接埠、使單一伺服器InfiniBand連接埠故障。
- 使用BMC觸發立即關閉伺服器電源。
- 將節點正常置於待命狀態、並將故障切換服務移轉至其他節點。
- 正常地將節點重新連線、並將服務容錯回復至原始節點。
- 使用PDU關閉其中一個InfiniBand交換器。所有測試都是在壓力測試進行期間執行、並在BeeGFS用戶端上設定「SysSessionChecksEnabled:假」參數。未發現I/O錯誤或中斷。

> 有已知問題（請參閱 "Changelog"）當BeeGFS用戶端/伺服器RDMA連線意外中斷時、可能是因為主要介面遺失（如「connInterfacesFile」中所定義）、或是BeeGFS伺服器故障；作用中用戶端I/O在恢復前最多可掛斷10分鐘。若BeeGFS節點在規劃維護時正常放置在待命或使用TCP、則不會發生此問題。

NVIDIA DGX SuperPOD 和 **BasePOD** 驗證

NetApp已使用類似的BeeGFS檔案系統（由三個建置區塊組成、並套用中繼資料加上儲存組態設定檔）、驗證NVIDIAs DGX A100 SupermPOD的儲存解決方案。此NVA所描述的解決方案、需要測試資格、測試20部DGX A100 GPU伺服器、執行各種儲存設備、機器學習和深度學習基準測試。以 NVIDIA DGX A100 SuperPOD 所建立的驗證為基礎、 NetApp 上的 BeeGFS 解決方案已獲得 DGX SuperPOD H100 、 H200 及 B200 系統的核准。這項延伸是根據 NVIDIA DGX A100 所驗證的先前基準測試和系統需求而定。

如需詳細資訊、請參閱 "NVIDIA DGX超級POD與NetApp合作" 和 "NVIDIA DGX基礎POD"。

規模調整準則

BeeGFS解決方案包含根據驗證測試來調整效能和容量規模的建議。

建置區塊架構的目標、是透過新增多個建置區塊來建立易於調整規模的解決方案、以符合特定BeeGFS系統的需

求。根據以下準則、您可以預估BeeGFS建置區塊的數量和類型、以符合您環境的需求。

請記住、這些預估是最佳的效能表現。綜合基準測試應用程式是以實際應用程式可能無法使用的方式來撰寫及使用、以最佳化基礎檔案系統的使用。

效能規模調整

下表提供建議的效能規模調整。

組態設定檔	**1MiB**讀取	**1MiB**寫入
中繼資料+儲存設備	62GiBps	21GiBps
僅儲存設備	64GiBps	21GiBps

中繼資料容量規模預估是根據「經驗法則」、在BeeGFS中、500 GB的容量足以容納約1.5億個檔案。（如需詳細資訊、請參閱BeeGFS文件 "系統需求"）

使用存取控制清單等功能、以及每個目錄的目錄和檔案數量、也會影響中繼資料空間的使用速度。儲存容量預估會考慮可用磁碟機容量、以及RAID 6和XFS負荷。

中繼資料**+**儲存建置區塊的容量規模

下表提供中繼資料與儲存建置區塊的建議容量規模調整。

磁碟機大小（**2+2 RAID 1**）中繼資料**Volume**群組	中繼資料容量（檔案數）	磁碟機大小（**8+2 RAID 6**）儲存**Volume**群組	儲存容量（檔案內容）
1.92TB	1,938,577,200	1.92TB	51.77TB
3.84 TB	3,880,388,400	3.84 TB	103.55TB
7.68TB	8、125、278、000	7.68TB	216.74 TB
15.3TB	17、269、854000	15.3TB	460.60TB

調整中繼資料加上儲存建置區塊規模時、您可以使用較小的磁碟機來進行中繼資料磁碟區群組、而非儲存磁碟區群組、藉此降低成本。

專為儲存設備建置區塊調整容量

下表針對純儲存建置區塊提供經驗法則容量規模調整。

磁碟機大小（**10+2 RAID 6**）儲存**Volume**群組	儲存容量（檔案內容）
1.92TB	59.89TB
3.84 TB	119.80TB
7.68TB	251.89TB
15.3TB	538.55TB

除非啟用全域檔案鎖定、否則在基礎（第一）建置區塊中納入管理服務的效能和容量負荷最小。

效能調校

BeeGFS解決方案包含根據驗證測試進行效能調校的建議。

儘管BeeGFS提供合理的開箱即用效能、但NetApp已開發出一套建議的調校參數、以最大化效能。這些參數會考量基礎E系列區塊節點的功能、以及在共享磁碟HA架構中執行BeeGFS所需的任何特殊需求。

檔案節點的效能調校

您可以設定的可用調校參數包括：

1. *檔案節點的UEFI/BIOS中的系統設定。*若要發揮最大效能、建議您在做為檔案節點的伺服器機型上設定系統設定。您可以使用系統設定程式（UEFI/BIOS）或底板管理控制器（BMC）提供的Redfish API來設定檔案節點時的系統設定。

   系統設定會因您用來做為檔案節點的伺服器機型而有所不同。這些設定必須根據使用中的伺服器機型手動設定。若要了解如何配置已驗證的 Lenovo SR665 V3 檔案節點的系統設置，請參閱"調整檔案節點系統設定以獲得效能" 。

2. *必要組態參數的預設設定。*必要的組態參數會影響BeeGFS服務的設定方式、以及E系列磁碟區（區塊裝置）如何由心臟起搏器設定格式及掛載。這些必要的組態參數包括：

   - BeeGFS服務組態參數

     您可以視需要覆寫組態參數的預設設定。如需可針對特定工作負載或使用案例進行調整的參數，請參閱 "BeeGFS服務組態參數"。

   - Volume格式化和掛載參數會設定為建議的預設值、而且只能針對進階使用案例進行調整。預設值會執行下列動作：

     - 根據目標類型（例如管理、中繼資料或儲存設備）、以及基礎磁碟區的RAID組態和區段大小、最佳化初始磁碟區格式。
     - 調整心臟起搏器如何掛載每個Volume、以確保變更立即排清至E系列區塊節點。如此可在檔案節點發生故障且正在進行作用中寫入時、避免資料遺失。

       如需可針對特定工作負載或使用案例進行調整的參數，請參閱 "Volume格式化與掛載組態參數"。

3. * 安裝在檔案節點上的 Linux 作業系統中的系統設定。 *當您在的步驟 4 中建立 Ansible 庫存時、您可以覆寫預設的 Linux 作業系統設定 "建立可Ansible庫存"。

   預設設定是用來驗證NetApp解決方案上的BeeGFS、但您可以變更這些設定、以因應您的特定工作負載或使用案例進行調整。您可以變更的一些Linux作業系統設定範例包括：

   - E系列區塊裝置上的I/O佇列。

     您可以在作為BeeGFS目標的E系列區塊裝置上設定I/O佇列、以便：

     - 根據裝置類型（NVMe、HDD等）調整排程演算法。
     - 增加未處理要求的數量。
     - 調整要求大小。
     - 最佳化預先讀取行為。

- 虛擬記憶體設定：

  您可以調整虛擬記憶體設定、以獲得最佳的持續串流效能。

- CPU設定：

  您可以調整CPU頻率調節器和其他CPU組態、以獲得最大效能。

- 讀取要求大小。

  您可以增加 NVIDIA HCA 的讀取要求大小上限。

區塊節點的效能調校

根據套用至特定BeeGFS建置區塊的組態設定檔、區塊節點上設定的Volume群組會稍微變更。例如、使用24個磁碟機EF600區塊節點：

- 對於單一基礎建置區塊、包括BeeGFS管理、中繼資料和儲存服務：
  - 1個2+2個RAID 10 Volume群組、用於BeeGFS管理和中繼資料服務
  - 2個8+2個RAID 6 Volume群組用於BeeGFS儲存服務
- 若為BeeGFS中繼資料+儲存建置區塊：
  - 1個2+2個RAID 10 Volume群組、用於BeeGFS中繼資料服務
  - 2個8+2個RAID 6 Volume群組用於BeeGFS儲存服務
- 僅適用於BeeGFS儲存設備建置區塊：
  - 2個10+2個RAID 6 Volume群組用於BeeGFS儲存服務

> 由於BeeGFS需要的管理與中繼資料儲存空間比儲存空間大幅減少、因此有一個選項是針對RAID 10 Volume群組使用較小的磁碟機。較小的磁碟機應安裝在最外側的磁碟機插槽中。如需詳細資訊、請參閱 "部署指示"。

這些都是由Ansible型部署所設定、以及其他一些一般建議的設定、以最佳化效能/行為、包括：

- 將全域快取區塊大小調整為32KiB、並將需求型快取排清調整為80%。
- 停用自動負載平衡（確保控制器磁碟區指派維持原定狀態）。
- 啟用讀取快取和停用預先讀取快取。
- 啟用含鏡射的寫入快取、並需要電池備份、以便在區塊節點控制器故障時、快取仍會持續存在。
- 指定磁碟機指派給磁碟區群組的順序、在可用磁碟機通道之間平衡I/O。

大容量建置區塊

標準BeeGFS解決方案設計以高效能工作負載為設計考量。尋求高容量使用案例的客戶應觀察此處概述的設計與效能特性差異。

硬體與軟體組態

高容量建置區塊的硬體和軟體組態為標準配置、但EF600控制器應更換為EF300控制器、並可選擇連接1到7個IOM擴充支架、每個儲存陣列各有60個磁碟機、每個建置區塊總計2至14個擴充托盤。

部署高容量建置區塊設計的客戶、可能只會使用由BeeGFS管理、中繼資料及每個節點儲存服務所組成的基礎建置區塊樣式組態。為了節省成本、大容量儲存節點應在EF300控制器機箱的NVMe磁碟上配置中繼資料磁碟區、並應將儲存磁碟區配置至擴充托盤中的NL-SAS磁碟機。

[]

規模調整準則

這些規模調整準則假設大容量建置區塊在基礎EF300機箱中設定一個2+2 NVMe SSD Volume群組作為中繼資料、並在每個IOM擴充匣中設定6x 8+2 NL-SAS Volume群組作為儲存設備。

磁碟機大小（容量**HDD**）	每個寬板的容量（**1**個紙匣）	每個寬帶容量（**2**個磁碟匣）	每個寬帶容量（**3**個磁碟匣）	每個寬帶容量（**4**個磁碟匣）
4TB	439TB	878 TB	1317 TB	1756 TB
8 TB	878 TB	1756 TB	2634 TB	3512 TB
10 TB	1097 TB	2195 TB	3292 TB	4390 TB
12 TB	1317 TB	2634 TB	3951 TB	5268TB
16 TB	1756 TB	3512 TB	5268TB	7024 TB
18 TB	1975 TB	3951 TB	5927 TB	7902 TB

部署解決方案

部署總覽

NetApp 上的 BeeGFS 可部署至已驗證的檔案和區塊節點，使用 Ansible 搭配 NetApp 的 BeeGFS 建置區塊設計。

Ansible集合與角色

NetApp 上的 BeeGFS 解決方案是使用 Ansible 部署， Ansible 是一款可自動化應用程式部署的熱門 IT 自動化引擎。Ansible 使用一系列的檔案，統稱為庫存，用來建立您要部署的 BeeGFS 檔案系統的模型。

Ansible 允許 NetApp 等公司使用 Ansible Galaxy 上的可用集合來擴充內建功能（請參閱 "NetApp E系列BeeGFS系列"）。集合包括執行特定功能或工作（例如建立 E 系列 Volume ）的模組，以及可呼叫多個模組和其他角色的角色。此自動化方法可縮短部署BeeGFS檔案系統和基礎HA叢集所需的時間。此外，它也簡化了叢集和 BeeGFS 檔案系統的維護與擴充。

如需其他詳細資料、請參閱 "瞭解Ansible庫存"。

由於在NetApp解決方案上部署BeeGFS涉及許多步驟、因此NetApp不支援手動部署解決方案。

BeeGFS建置區塊的組態設定檔

部署程序涵蓋下列組態設定檔：

- 一個基礎建置區塊、包含管理、中繼資料和儲存服務。
- 第二個建置區塊、包含中繼資料和儲存服務。
- 僅包含儲存服務的第三個建置區塊。

這些設定檔說明NetApp BeeGFS建置區塊的完整建議組態設定檔。對於每個部署，中繼資料和儲存建置區塊或僅儲存服務的建置區塊數量可能會因容量和效能需求而異。

部署步驟總覽

部署作業包括下列高層級工作：

硬體部署

1. 實際組裝每個建置區塊。
2. 機架與纜線硬體。如需詳細程序、請參閱 "部署硬體"。

軟體部署

1. "設定檔案和區塊節點"。
   - 在檔案節點上設定BMC IP
   - 安裝支援的作業系統、並在檔案節點上設定管理網路
   - 在區塊節點上設定管理IP
2. "設定可Ansible控制節點"。
3. "調整系統設定以獲得效能"。
4. "建立可Ansible庫存"。
5. "定義BeeGFS建置區塊的Ansible庫存"。
6. "使用Ansible部署BeeGFS"。
7. "設定BeeGFS用戶端"。

部署程序包括幾個需要將文字複製到檔案的範例。密切注意任何以“#”或“//”字元表示的內聯註釋，其中的內容應該或可以針對特定部署進行修改。例如：

```
`beegfs_ha_ntp_server_pools:  # THIS IS AN EXAMPLE OF A COMMENT!
  - "pool 0.pool.ntp.org iburst maxsources 3"
  - "pool 1.pool.ntp.org iburst maxsources 3"`
```

衍生架構、部署建議有多種差異：

- "高容量建置區塊"

瞭解Ansible庫存

在開始部署之前，請先熟悉 Ansible 的設定方式，以及如何在 NetApp 解決方案上部署 BeeGFS 。

Ansible 清單是一個目錄結構，列出要跨部署 BeeGFS 檔案系統的檔案和區塊節點。其中包含主機，群組和變數，說明所需的 BeeGFS 檔案系統。Ansible 庫存必須儲存在 Ansible 控制節點上，而 Ansible 控制節點是可存取檔案的任何機器，以及用來執行 Ansible 教戰手冊的區塊節點。您可以從下載範例庫存 "NetApp E系列BeeGFS GitHub"。

Ansible模組與角色

若要套用 Ansible 庫存所述的組態，請使用 NetApp E 系列 Ansible 集合（可從取得）中提供的各種 Ansible 模組和角色 "NetApp E系列BeeGFS GitHub"，以部署端點對端點解決方案。

NetApp E系列Ansible產品組合中的每個角色、都是完整的BeeGFS on NetApp解決方案端點對端部署。這些角色使用NetApp E系列SANtricity 的《Sf2、Host和BeeGFS》集合、可讓您使用HA（高可用度）來設定BeeGFS檔案系統。然後您可以配置及對應儲存設備、並確保叢集儲存設備已準備就緒可供使用。

雖然角色會提供深入的文件、但部署程序會說明如何使用第二代BeeGFS建置區塊設計來部署NetApp驗證架構。

> ⓘ 雖然部署步驟會嘗試提供足夠的詳細資料、以確保事先使用Ansible的經驗並非先決條件、但您應該對Ansible及相關術語有一定的瞭解。

BeeGFS HA叢集的庫存配置

使用 Ansible 庫存結構定義 BeeGFS HA 叢集。

任何具有前一次 Ansible 體驗的人都應該知道 BeeGFS HA 角色會實作自訂方法，以探索哪些變數（或事實）適用於每個主機。這項設計簡化了 Ansible 庫存的結構，以描述可在多部伺服器上執行的資源。

Ansible 庫存通常包含和 `group_vars`中的檔案 `host_vars`，以及 `inventory.yml`將主機指派給特定群組的檔案（以及可能群組至其他群組的檔案）。

> ⓘ 請勿使用本小節中的內容建立任何檔案、僅供範例使用。

雖然此組態是根據組態設定檔預先決定的、但您應該大致瞭解如何將所有項目設定為「可執行」清單、如下所示：

```
# BeeGFS HA (High Availability) cluster inventory.
all:
  children:
    # Ansible group representing all block nodes:
    eseries_storage_systems:
      hosts:
        netapp01:
        netapp02:
    # Ansible group representing all file nodes:
    ha_cluster:
      children:
        meta_01:  # Group representing a metadata service with ID 01.
          hosts:
            beegfs_01:  # This service is preferred on the first file
node.
            beegfs_02:  # And can failover to the second file node.
        meta_02:  # Group representing a metadata service with ID 02.
          hosts:
            beegfs_02:  # This service is preferred on the second file
node.
            beegfs_01: # And can failover to the first file node.
```

對於每項服務、會在「group _vars」下建立一個額外的檔案、說明其組態：

```
# meta_01 - BeeGFS HA Metadata Resource Group
beegfs_ha_beegfs_meta_conf_resource_group_options:
  connMetaPortTCP: 8015
  connMetaPortUDP: 8015
  tuneBindToNumaZone: 0
floating_ips:
  - i1b: <IP>/<SUBNET_MASK>
  - i2b: <IP>/<SUBNET_MASK>
# Type of BeeGFS service the HA resource group will manage.
beegfs_service: metadata # Choices: management, metadata, storage.
# What block node should be used to create a volume for this service:
beegfs_targets:
  netapp01:
    eseries_storage_pool_configuration:
      - name: beegfs_m1_m2_m5_m6
        raid_level: raid1
        criteria_drive_count: 4
        common_volume_configuration:
          segment_size_kb: 128
        volumes:
          - size: 21.25
            owning_controller: A
```

此配置可讓每個資源的BeeGFS服務、網路和儲存組態在單一位置定義。在幕後、BeeGFS角色會根據此庫存結構、針對每個檔案和區塊節點集合必要的組態。

每項服務的BeeGFS數字和字串節點ID會根據群組名稱自動設定。因此、除了群組名稱必須是唯一的一般「可獨立」要求之外、代表BeeGFS服務的群組必須以該群組所代表之BeeGFS服務類型的唯一數字結尾。例如、中繼資料_01和stOR_01是允許的、但中繼資料_01和meta_01則不允許。

檢視最佳實務做法

在NetApp解決方案上部署BeeGFS時、請遵循最佳實務準則。

標準慣例

實際組裝及建立Ansible庫存檔案時、請遵循下列標準慣例（如需詳細資訊、請參閱） "建立可Ansible庫存"）。

- 檔案節點主機名稱會依序編號（H01-HN）、機架頂端的數字較低、底部的數字較高。

  例如，命名慣例 `[location][row][rack]hN` 如下所示： `beegfs_01`。

- 每個區塊節點都由兩個儲存控制器組成、每個控制器都有自己的主機名稱。

  儲存陣列名稱是指可Ansible庫存中的整個區塊儲存系統。儲存陣列名稱應依序編號（A01-A）、個別控制器的主機名稱則衍生自該命名慣例。

例如，通常名稱為的區塊節點 `ictad22a01` 可以為每個控制器設定主機名稱，例如 `ictad22a01-a` 和 `ictad22a01-b`，但在 Ansible 清單中則稱為 `netapp_01`。

- 同一個建置區塊內的檔案和區塊節點共用相同的編號配置、並在機架中彼此相鄰、兩個檔案節點位於頂端、兩個區塊節點位於其正下方。

  例如、在第一個建置區塊中、檔案節點H01和h02都直接連接至區塊節點A01和A02。從上到下、主機名稱為H01、h02、A01和A02。

- 建置區塊會根據主機名稱以連續順序安裝、因此編號較低的主機名稱位於機架頂端、編號較高的主機名稱位於底部。

  其目的是將連接至機架交換器頂端的纜線長度降至最低、並定義標準部署實務做法、以簡化疑難排解。如果資料中心因為擔心機架穩定性而不允許使用此功能、則肯定會允許使用相反的功能、從底部向上填入機架。

InfiniBand儲存網路組態

每個檔案節點上的一半InfiniBand連接埠、用於直接連線至區塊節點。另一半連接至InfiniBand交換器、用於BeeGFS用戶端與伺服器的連線。在判斷用於BeeGFS用戶端和伺服器的IPoIB子網路大小時、您必須考量運算/GPU叢集和BeeGFS檔案系統的預期成長。如果您必須偏離建議的IP範圍、請記住、單一建置區塊中的每個直接連線都有獨特的子網路、而且不會與用於用戶端與伺服器連線的子網路重疊。

直接連線

每個建置區塊內的檔案和區塊節點、一律使用下表中的IP進行直接連線。

此定址方案遵循下列規則：第三個八位元組永遠是不規則的、甚至是不規則的、這取決於檔案節點是不規則的或是偶數的。

檔案節點	**IB**連接埠	**IP** 位址	區塊節點	**IB**連接埠	實體**IP**	虛擬**IP**
ODD（上一）	i1a.	192.168.1.10	ODD（C1）	2A.	192.168.1.100	192.168.1.101
ODD（上一）	I2A	192.168.1.10	ODD（C1）	2A.	192.168.3.100	192.168.3.101
ODD（上一）	i3a	192.168.5.10	偶數（C2）	2A.	192.168.5.100	192.168.5.101
ODD（上一）	i4a.	192.168.1.10	偶數（C2）	2A.	192.168.1.100	192.168.1.101
偶數（下半年）	i1a.	192.168.1.10	ODD（C1）	2B	192・168・2・100	192・168・2・101
偶數（下半年）	I2A	192・168・4・10	ODD（C1）	2B	24.100	24.101
偶數（下半年）	i3a	地址：192・168・6・10	偶數（C2）	2B	6.100	6.101
偶數（下半年）	i4a.	192・168・8・10	偶數（C2）	2B	192・168・8・100	192・168・8・101

BeeGFS 用戶端伺服器 IPoIB 定址方案

每個檔案節點都會執行多個BeeGFS伺服器服務（管理、中繼資料或儲存設備）。為了讓每項服務獨立容錯移轉至其他檔案節點、每項服務都會設定獨特的IP位址、以便在兩個節點之間浮動（有時稱為邏輯介面或LIF）。

此部署雖然並非必要、但會假設這些連線使用下列IPoIB子網路範圍、並定義套用下列規則的標準定址方案：

- 第二個八位元組永遠是不符合或甚至不符合、取決於檔案節點InfiniBand連接埠是ODD或偶數。
- BeeGFS叢集IP永遠是「xxx」。127.100.yyy'或'xxx.xxx.128.100.y'。

> 除了用於頻內作業系統管理的介面、電暈同步還能使用其他介面來進行叢集心律跳轉和同步。如此可確保單一介面遺失不會導致整個叢集中斷運作。

- BeeGFS管理服務永遠是「xxx.xxx.Y.101.0」或「xxx.xxx.Y.102.0」。
- BeeGFS中繼資料服務一律位於「xxx.yyy.101.zzz」或「xxx.xxx.y.102.zzz」。
- BeeGFS 儲存服務永遠位於 `xxx.yyy.103.zzz` 或 `xxx.yyy.104.zzz`。
- 範圍從「100.xxx.1.1」到「100.xxx.99.255」的位址會保留給用戶端。

IPoIB 單一子網路定址方案

根據中列出的優點，本部署指南將使用單一子網路架構 "軟體架構"。

子網路： **100.127.0.0/16**
下表提供單一子網路的範圍： 100.127.0.0.0/16 。

目的	**InfiniBand**連接埠	**IP**位址或範圍
BeeGFS叢集IP	i1b 或 i4b	100127.100.1 - 100127.1005.255
BeeGFS管理	i1b	100127.101.0
	i2b	100.127.102.0
BeeGFS中繼資料	i1b或i3b	100127.101.1 - 100127.101.255
	i2b或i4b	100.127.102.1 - 100.127.102.255
BeeGFS儲存設備	i1b或i3b	100127.103.1 - 100127.103.255
	i2b或i4b	100.127.104.1-100.127.104.255
BeeGFS用戶端	（因用戶端而異）	100127.1.1 - 100127.99.255

IPoIB 兩個子網路定址方案

不再建議使用兩個子網路定址方案、但仍可實作。如需建議的兩個子網路配置、請參閱下表。

子網路**A**：**100127.0/16**
下表提供子網路A的範圍：100127.0.0/16。

目的	**InfiniBand**連接埠	**IP**位址或範圍
BeeGFS叢集IP	i1b	100127.100.1 - 100127.1005.255
BeeGFS管理	i1b	100127.101.0
BeeGFS中繼資料	i1b或i3b	100127.101.1 - 100127.101.255
BeeGFS儲存設備	i1b或i3b	100127.103.1 - 100127.103.255

目的	**InfiniBand**連接埠	**IP**位址或範圍
BeeGFS用戶端	（因用戶端而異）	100127.1.1 - 100127.99.255

子網路**B**：**100128.0/16**

下表提供子網路B的範圍：100128.0.0/16。

目的	**InfiniBand**連接埠	**IP**位址或範圍
BeeGFS叢集IP	i4b.	100128.100.1 - 100128.1005.255
BeeGFS管理	i2b	100128.102.0
BeeGFS中繼資料	i2b或i4b	100128.102.1 - 100128.102.255
BeeGFS儲存設備	i2b或i4b	100128.104.1 - 100128.104.255
BeeGFS用戶端	（因用戶端而異）	100128.1.1 - 100128.99.255

並非上述範圍內的所有IP都用於此NetApp認證架構。它們示範如何預先配置IP位址、以便使用一致的IP定址方案輕鬆擴充檔案系統。在此方案中、BeeGFS檔案節點和服務ID對應於已知IP範圍的第四個八位元組。如果需要、檔案系統當然可以擴充至超過255個節點或服務。

部署硬體

每個建置區塊都包含兩個已驗證的x86檔案節點、這些節點使用HDR（200GB）InfiniBand纜線直接連接至兩個區塊節點。

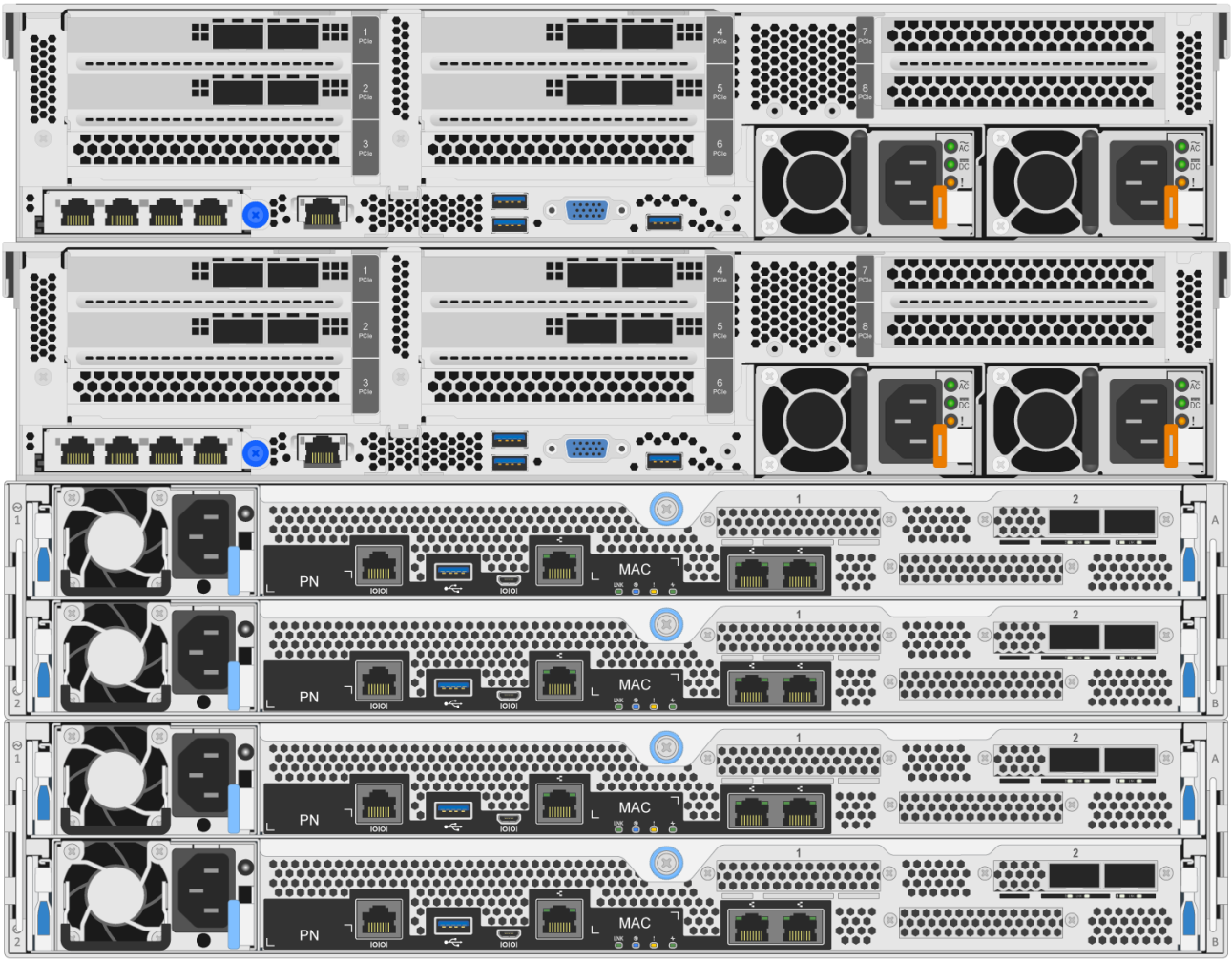
在容錯移轉叢集中建立仲裁所需的建置區塊至少有兩個。雙節點叢集具有可能會阻止容錯移轉成功的限制。您可以將第三個裝置整合為 tiebreaker 來設定雙節點叢集、但本文件並未說明該設計。

下列步驟對於叢集中的每個建置區塊都是相同的、無論是用於執行 BeeGFS 中繼資料和儲存服務、還是僅用於儲存服務、除非另有說明。

步驟

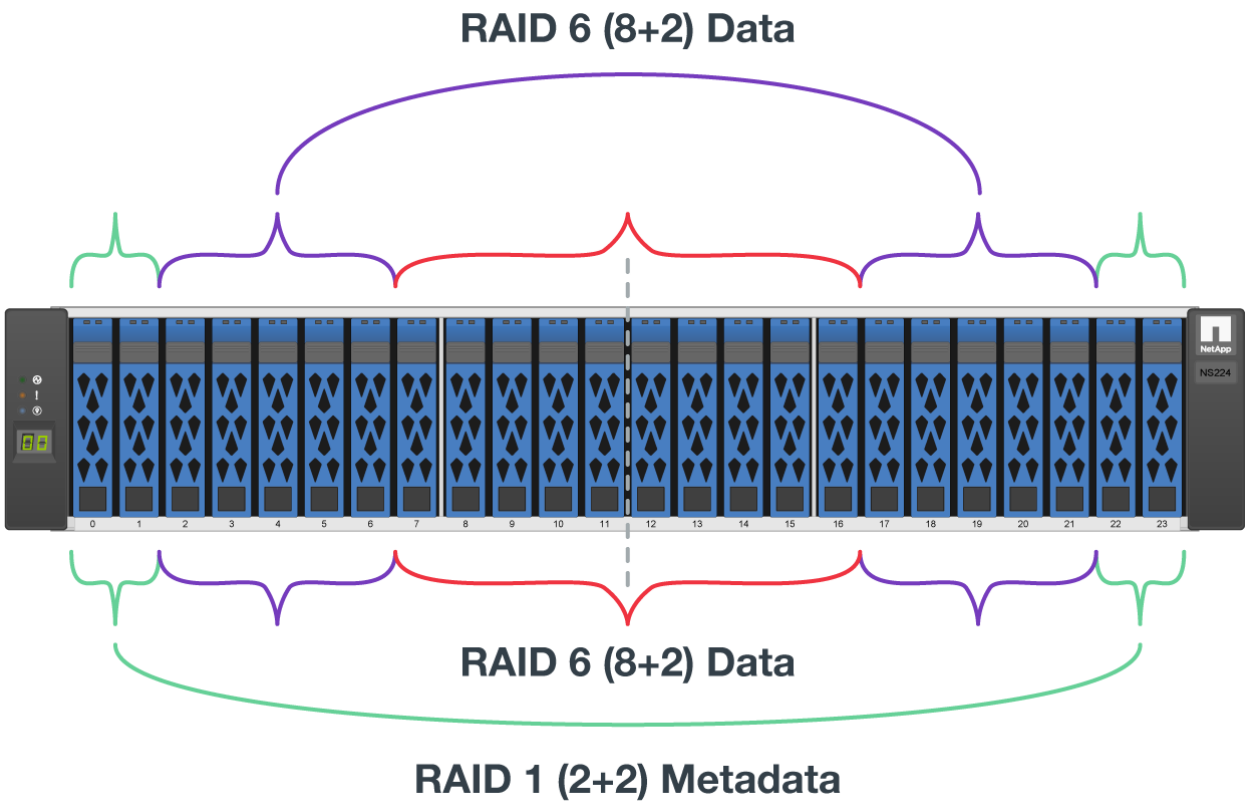
1. 使用中指定的模型，設定每個 BeeGFS 檔案節點與四個主機通道介面卡 (HCAs) "技術需求"。根據下列規格、將 HCA 插入檔案節點的 PCIe 插槽：
   - * Lenovo ThinkSystem SR665 V3 伺服器： * 使用 PCIe 插槽 1 、 2 、 4 和 5 。
   - * Lenovo ThinkSystem SR665 伺服器： * 使用 PCIe 插槽 2 、 3 、 5 和 6 。
2. 使用雙埠200GB主機介面卡（HIC）設定每個BeeGFS區塊節點、並在其兩個儲存控制器中的每個都安裝HIC。

   將建置區塊架起、使兩個BeeGFS檔案節點在BeeGFS區塊節點上方。下圖顯示 BeeGFS 建置區塊的正確硬體組態、使用 Lenovo ThinkSystem SR665 V3 伺服器做為檔案節點（後視圖）。

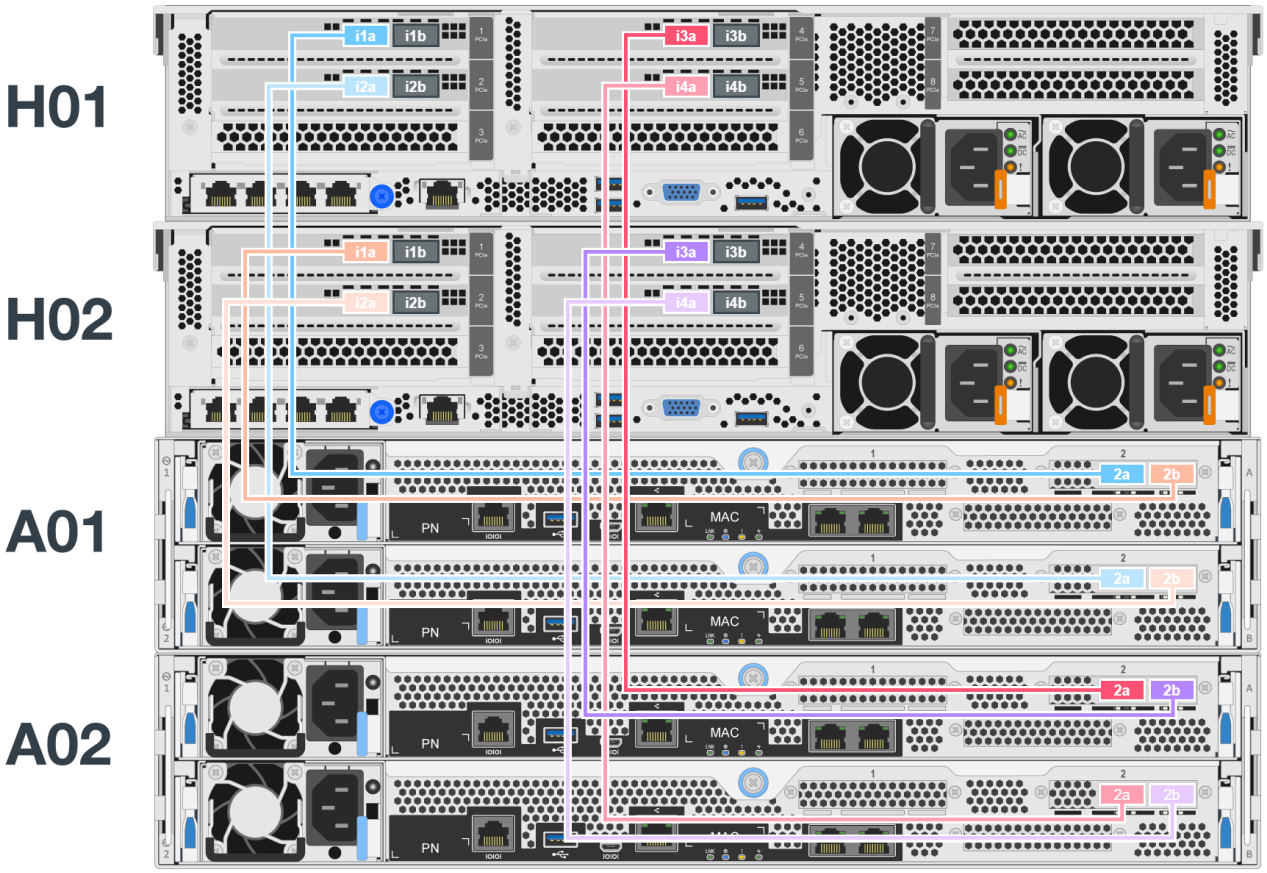


生產使用案例的電源供應器組態通常應使用備援PSU。

3. 如有需要、請在每個BeeGFS區塊節點中安裝磁碟機。
   a. 如果建置區塊將用於執行BeeGFS中繼資料和儲存服務、而較小的磁碟機則用於中繼資料磁碟區、請確認它們已安裝在最外側的磁碟機插槽中、如下圖所示。
   b. 對於所有的建置區塊組態、如果磁碟機機箱未完全安裝、請確定插槽0–11和12–23中已安裝相同數量的磁碟機、以獲得最佳效能。



4. 使用連接區塊和檔案節點 "1M InfiniBand HDR 200GB 直接連接銅線"、使它們符合下圖所示的拓撲。

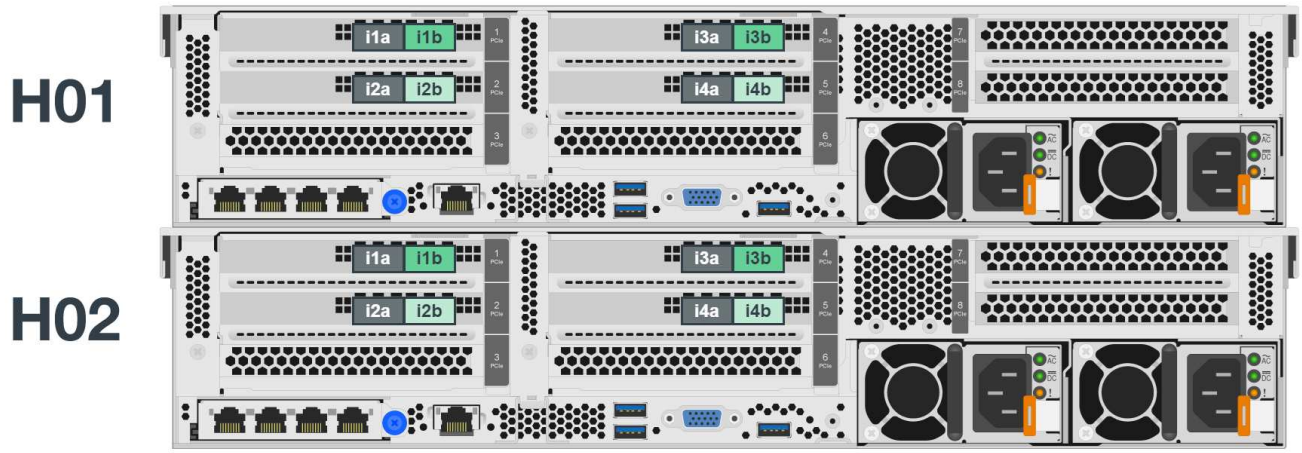


橫跨多個建置區塊的節點永遠不會直接連線。每個建置區塊都應視為獨立式單元、而建置區塊之間的所有通訊都是透過網路交換器進行。

5. 使用特定於 InfiniBand 儲存交換器的、將檔案節點上的其餘 InfiniBand 連接埠連接至儲存網路的 InfiniBand 交換 "2 公尺 InfiniBand 纜線" 器。

   當使用分離器纜線將儲存交換器連接至檔案節點時、一條纜線應從交換器分支出來、並連接至淡綠色的連接埠。另一條分離器纜線應從交換器分支出來、並連接至暗綠色的連接埠。

   此外、對於具有備援交換器的儲存網路、淡綠色的連接埠應連接至一台交換器、而深綠色的連接埠則應連接至另一台交換器。



6. 視需要、依照相同的佈線準則組裝其他建置組塊。

可部署在單一機架中的建置區塊總數、取決於每個站台可用的電力和冷卻。

部署軟體

設定檔案節點和區塊節點

雖然大部分的軟體組態工作都是使用NetApp提供的Ansible集合來自動化、但您必須在每部伺服器的底板管理控制器（BMC）上設定網路、並在每個控制器上設定管理連接埠。

設定檔案節點

1. 在每部伺服器的基礎板管理控制器（BMC）上設定網路。

   若要瞭解如何為已驗證的 Lenovo SR665 V3 檔案節點設定網路、請參閱 "Lenovo ThinkSystem文件"。

底板管理控制器（BMC）有時稱為服務處理器、是內建於各種伺服器平台的頻外管理功能的一般名稱、即使作業系統未安裝或無法存取、也能提供遠端存取。廠商通常會以自己的品牌行銷這項功能。例如、在Lenovo SR665上、BMC稱為_Lenovo XClarity Controller（XCC）_。

2. 設定系統設定以獲得最大效能。

您可以使用UEFI設定（先前稱為BIOS）或使用許多BMC提供的Redfish API來設定系統設定。系統設定會因做為檔案節點的伺服器機型而有所不同。

若要了解如何配置已驗證的 Lenovo SR665 V3 檔案節點的系統設置，請參閱"調整系統設定以獲得效能" 。

3. 安裝 Red Hat Enterprise Linux (RHEL) 9.4 並設定用於管理作業系統的主機名稱和網路端口，包括來自 Ansible 控制節點的 SSH 連線。

   此時請勿在任何InfiniBand連接埠上設定IP。

   > 雖然並非嚴格要求、但後續章節假設主機名稱會依序編號（例如H1-HN）、並提及應該在ODD或偶數主機上完成的工作。

4. 使用 Red Hat Subscription Manager 註冊並訂閱系統，以允許從官方 Red Hat 儲存庫安裝所需的軟體包，並將更新限制在支援的 Red Hat 版本上： `subscription-manager release --set=9.4` 。有關說明，請參閱 "如何註冊及訂閱RHEL系統" 和 "如何限制更新"。

5. 啟用包含高可用度所需套件的Red Hat儲存庫。

   ```
   subscription-manager repo-override --repo=rhel-9-for-x86_64
   -highavailability-rpms --add=enabled:1
   ```

6. 將所有 HCA 韌體更新至使用"更新檔案節點介面卡韌體"指南中建議的版本"技術需求"。

設定區塊節點

設定每個控制器上的管理連接埠、以設定EF600區塊節點。

1. 在每個EF600控制器上設定管理連接埠。

   有關配置端口的說明，請轉至 "E系列文件中心"。

2. （可選）設定每個系統的儲存陣列名稱。

   設定名稱可讓您更容易在後續章節中參考每個系統。有關設置陣列名稱的說明，請轉至 "E系列文件中心"。

> 雖然並非嚴格要求、但後續主題會假設儲存陣列名稱會依序編號（例如C1 - CN）、並提及應該在ODD或偶數系統上完成的步驟。

調整檔案節點系統設定以獲得效能

若要發揮最大效能、建議您在您做為檔案節點的伺服器機型上設定系統設定。

系統設定會因您用來做為檔案節點的伺服器機型而有所不同。本主題說明如何設定已驗證之Lenovo ThinkSystem SR665伺服器檔案節點的系統設定。

使用**UEFI**介面調整系統設定

Lenovo SR665 V3 伺服器的系統韌體包含許多可透過 UEFI 介面設定的調整參數。這些調校參數可能會影響伺服器運作方式的所有層面、以及伺服器的效能表現。

在* UEFI Setup > System Settings*下、調整下列系統設定：

操作模式功能表

系統設定	變更為
操作模式	自訂
CTDP	手冊
CTDP手冊	350
套件電力限制	手冊
效率模式	停用
globe-C態 控制	停用
SOC P狀態	P0
DF C狀態	停用
P-State	停用
啟用記憶體關機	停用
每個插槽的NUMA節點	NPS1

裝置和**I/O**連接埠功能表

系統設定	變更為
IOMMU	停用

電源選單

系統設定	變更為
PCIe Power Brake	停用

處理器功能表

系統設定	變更為
全域C狀態控制	停用
DF C狀態	停用
SMT模式	停用
CPPC	停用

使用**Redfish API**調整系統設定

除了使用UEFI設定、您也可以使用Redfish API來變更系統設定。

```
curl --request PATCH \
  --url https://<BMC_IP_ADDRESS>/redfish/v1/Systems/1/Bios/Pending \
  --user <BMC_USER>:<BMC- PASSWORD> \
  --header 'Content-Type: application/json' \
  --data '{
"Attributes": {
"OperatingModes_ChooseOperatingMode": "CustomMode",
"Processors_cTDP": "Manual",
"Processors_PackagePowerLimit": "Manual",
"Power_EfficiencyMode": "Disable",
"Processors_GlobalC_stateControl": "Disable",
"Processors_SOCP_states": "P0",
"Processors_DFC_States": "Disable",
"Processors_P_State": "Disable",
"Memory_MemoryPowerDownEnable": "Disable",
"DevicesandIOPorts_IOMMU": "Disable",
"Power_PCIePowerBrake": "Disable",
"Processors_GlobalC_stateControl": "Disable",
"Processors_DFC_States": "Disable",
"Processors_SMTMode": "Disable",
"Processors_CPPC": "Disable",
"Memory_NUMANodesperSocket":"NPS1"
}
}
'
```

如需Redfish架構的詳細資訊、請參閱 "DMTF網站"。

設定可**Ansible**控制節點

若要設定 Ansible 控制節點，您必須指定可透過網路存取的虛擬或實體機器，以存取

NetApp 解決方案上 BeeGFS 部署的所有檔案和區塊節點。

請檢閱"技術需求"以取得建議套件版本的清單。下列步驟已在 Ubuntu 22.04 上測試。有關首選 Linux 發行套件的特定步驟，請參閱 "Ansible文件"。

1. 從 Ansible 控制節點安裝下列 Python 和 Python Virtual Environment 套件。

```
sudo apt-get install python3 python3-pip python3-setuptools python3.10-venv
```

2. 建立 Python 虛擬環境。

```
python3 -m venv ~/pyenv
```

3. 啟動虛擬環境。

```
source ~/pyenv/bin/activate
```

4. 在啟動的虛擬環境中安裝所需的 Python 套件。

```
pip install ansible netaddr cryptography passlib
```

5. 使用 Ansible Galaxy 安裝 BeeGFS 集合。

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

6. 驗證 Ansible ， Python 和 BeeGFS 集合的安裝版本是否與匹配"技術需求"。

```
ansible --version
ansible-galaxy collection list netapp_eseries.beegfs
```

7. 設定無密碼 SSH ，允許 Ansible 從 Ansible 控制節點存取遠端 BeeGFS 檔案節點。
   a. 如果需要，在 Ansible 控制節點上產生一對公開金鑰。

```
ssh-keygen
```

   b. 為每個檔案節點設定無密碼 SSH 。

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

請*不要*設定區塊節點的無密碼SSH。這既不受支援、也不需要。

建立可**Ansible**庫存

若要定義檔案和區塊節點的組態、您可以建立可執行的詳細目錄、以代表您要部署的BeeGFS檔案系統。清單包括主機、群組和變數、說明所需的BeeGFS檔案系統。

步驟**1**：定義所有建置區塊的組態

定義套用至所有建置區塊的組態、無論您個別套用至哪些組態設定檔。

開始之前

- 為您的部署選擇子網路定址方案。由於中列出的優點 "軟體架構"、建議您使用單一子網路定址方案。

步驟

1. 在Ansible控制節點上、找出您要用來儲存Ansible庫存和教戰手冊檔案的目錄。

   除非另有說明、否則會針對此目錄建立此步驟中建立的所有檔案和目錄、以及執行下列步驟。

2. 建立下列子目錄：

   《host_vars》

   《團體》

   "套裝軟體"

3. 建立叢集密碼的子目錄，並使用 Ansible Vault 加密檔案以保護檔案安全（請參閱 "使用Ansible Vault加密內容"）：

   a. 創建子目錄 `group_vars/all`。

   b. 在 `group_vars/all`目錄中，建立一個標示為的密碼檔案 `passwords.yml`。

   c. 使用下列項目填入 `passwords.yml file`，根據您的組態取代所有的使用者名稱和密碼參數：

```
# Credentials for storage system's admin password
eseries_password: <PASSWORD>

# Credentials for BeeGFS file nodes
ssh_ha_user: <USERNAME>
ssh_ha_become_pass: <PASSWORD>

# Credentials for HA cluster
ha_cluster_username: <USERNAME>
ha_cluster_password: <PASSWORD>
ha_cluster_password_sha512_salt: randomSalt

# Credentials for fencing agents
# OPTION 1: If using APC Power Distribution Units (PDUs) for fencing:
# Credentials for APC PDUs.
apc_username: <USERNAME>
apc_password: <PASSWORD>

# OPTION 2: If using the Redfish APIs provided by the Lenovo XCC (and
other BMCs) for fencing:
# Credentials for XCC/BMC of BeeGFS file nodes
bmc_username: <USERNAME>
bmc_password: <PASSWORD>
```

d. 在出現提示時執行 `ansible-vault encrypt passwords.yml`並設定資料保險箱密碼。

步驟**2**：定義個別檔案和區塊節點的組態

定義適用於個別檔案節點和個別建置區塊節點的組態。

1. 在「host_vars/」下、為每個BeeGFS檔案節點建立一個名為「.yml」的檔案、其中包含下列內容、並特別注意BeeGFS叢集IP和主機名稱的填入內容、這些內容以odd或偶數結尾。

   一開始、檔案節點介面名稱會與此處列出的名稱相符（例如ib0或ibs1f0）。這些自訂名稱是在中設定 [步驟4：定義應套用至所有檔案節點的組態]。

```
ansible_host: "<MANAGEMENT_IP>"
eseries_ipoib_interfaces:  # Used to configure BeeGFS cluster IP
addresses.
  - name: i1b
    address: 100.127.100. <NUMBER_FROM_HOSTNAME>/16
  - name: i4b
    address: 100.127.100. <NUMBER_FROM_HOSTNAME>/16
beegfs_ha_cluster_node_ips:
  - <MANAGEMENT_IP>
  - <i1b_BEEGFS_CLUSTER_IP>
  - <i4b_BEEGFS_CLUSTER_IP>
# NVMe over InfiniBand storage communication protocol information
# For odd numbered file nodes (i.e., h01, h03, ..):
eseries_nvme_ib_interfaces:
  - name: i1a
    address: 192.168.1.10/24
    configure: true
  - name: i2a
    address: 192.168.3.10/24
    configure: true
  - name: i3a
    address: 192.168.5.10/24
    configure: true
  - name: i4a
    address: 192.168.7.10/24
    configure: true
# For even numbered file nodes (i.e., h02, h04, ..):
# NVMe over InfiniBand storage communication protocol information
eseries_nvme_ib_interfaces:
  - name: i1a
    address: 192.168.2.10/24
    configure: true
  - name: i2a
    address: 192.168.4.10/24
    configure: true
  - name: i3a
    address: 192.168.6.10/24
    configure: true
  - name: i4a
    address: 192.168.8.10/24
    configure: true
```

如果您已經部署BeeGFS叢集、則必須先停止叢集、再新增或變更靜態設定的IP位址、包括用於NVMe/IB的叢集IP和IP。這是必要的、因此這些變更會正常生效、而且不會中斷叢集作業。

2. 在「host_vars/」下、為每個BeeGFS區塊節點建立一個名為「<主機名稱>.yml」的檔案、然後填入下列內容。

   請特別注意要填入以odd結尾的儲存陣列名稱與偶數結尾之內容的相關注意事項。

   針對每個區塊節點、建立一個檔案、然後為兩個控制器之一（通常為A）指定「<management _ip>」（管理IP）。

```
eseries_system_name: <STORAGE_ARRAY_NAME>
eseries_system_api_url: https://<MANAGEMENT_IP>:8443/devmgr/v2/
eseries_initiator_protocol: nvme_ib
# For odd numbered block nodes (i.e., a01, a03, ..):
eseries_controller_nvme_ib_port:
  controller_a:
    - 192.168.1.101
    - 192.168.2.101
    - 192.168.1.100
    - 192.168.2.100
  controller_b:
    - 192.168.3.101
    - 192.168.4.101
    - 192.168.3.100
    - 192.168.4.100
# For even numbered block nodes (i.e., a02, a04, ..):
eseries_controller_nvme_ib_port:
  controller_a:
    - 192.168.5.101
    - 192.168.6.101
    - 192.168.5.100
    - 192.168.6.100
  controller_b:
    - 192.168.7.101
    - 192.168.8.101
    - 192.168.7.100
    - 192.168.8.100
```

步驟**3**：定義應套用至所有檔案和區塊節點的組態

您可以在與群組對應的檔案名稱中、定義「group _vars」下一組主機的通用組態。如此可避免在多個位置重複執行共用組態。

關於這項工作

主機可以位於多個群組中、執行時、Ansible會根據其可變優先順序規則、選擇要套用到特定主機的變數。（如需這些規則的詳細資訊、請參閱的「Ansible」文件 "使用變數"）

主機對群組指派是在實際的Ansible庫存檔案中定義、此檔案是在本程序結束時建立的。

步驟

在Ansible中、您想要套用至所有主機的任何組態都可以定義為「All（全部）」群組。使用下列內容建立檔案「group_vars/all.yml」：

```
ansible_python_interpreter: /usr/bin/python3
beegfs_ha_ntp_server_pools:  # Modify the NTP server addressess if
desired.
  - "pool 0.pool.ntp.org iburst maxsources 3"
  - "pool 1.pool.ntp.org iburst maxsources 3"
```

步驟**4**：定義應套用至所有檔案節點的組態

檔案節點的共用組態是在稱為「ha_cluster」的群組中定義。本節中的步驟會建置應包含在「group vars/ha_cluster・yml」檔案中的組態。

步驟

1. 在檔案頂端、定義預設值、包括在檔案節點上用做「show」使用者的密碼。

```
### ha_cluster Ansible group inventory file.
# Place all default/common variables for BeeGFS HA cluster resources
below.
### Cluster node defaults
ansible_ssh_user: {{ ssh_ha_user }}
ansible_become_password: {{ ssh_ha_become_pass }}
eseries_ipoib_default_hook_templates:
  - 99-multihoming.j2   # This is required for single subnet
deployments, where static IPs containing multiple IB ports are in the
same IPoIB subnet. i.e: cluster IPs, multirail, single subnet, etc.
# If the following options are specified, then Ansible will
automatically reboot nodes when necessary for changes to take effect:
eseries_common_allow_host_reboot: true
eseries_common_reboot_test_command: "! systemctl status
eseries_nvme_ib.service || systemctl --state=exited | grep
eseries_nvme_ib.service"
eseries_ib_opensm_options:
  virt_enabled: "2"
  virt_max_ports_in_process: "0"
```

> 如果 `ansible_ssh_user`已經 `root`是，則您可以選擇性地省略，`ansible_become_password`並在執行教戰手冊時指定 `--ask-become-pass`選項。

2. 您也可以設定高可用度（HA）叢集的名稱、並指定叢集內通訊的使用者。

   如果您要修改私有IP定址方案、也必須更新預設的「beegfs_ha_mgmtd_浮 點IP」。這必須符合您稍後為BeeGFS管理資源群組所設定的項目。

使用「beegfs_ha_alert_email_lists」指定一封或多封應接收叢集事件警示的電子郵件。

```
### Cluster information
beegfs_ha_firewall_configure: True
eseries_beegfs_ha_disable_selinux: True
eseries_selinux_state: disabled
# The following variables should be adjusted depending on the desired
configuration:
beegfs_ha_cluster_name: hacluster                    # BeeGFS HA cluster
name.
beegfs_ha_cluster_username: "{{ ha_cluster_username }}" # Parameter for
BeeGFS HA cluster username in the passwords file.
beegfs_ha_cluster_password: "{{ ha_cluster_password }}" # Parameter for
BeeGFS HA cluster username's password in the passwords file.
beegfs_ha_cluster_password_sha512_salt: "{{
ha_cluster_password_sha512_salt }}" # Parameter for BeeGFS HA cluster
username's password salt in the passwords file.
beegfs_ha_mgmtd_floating_ip: 100.127.101.0          # BeeGFS management
service IP address.
# Email Alerts Configuration
beegfs_ha_enable_alerts: True
beegfs_ha_alert_email_list: ["email@example.com"]  # E-mail recipient
list for notifications when BeeGFS HA resources change or fail.  Often a
distribution list for the team responsible for managing the cluster.
beegfs_ha_alert_conf_ha_group_options:
      mydomain: “example.com”
# The mydomain parameter specifies the local internet domain name. This
is optional when the cluster nodes have fully qualified hostnames (i.e.
host.example.com).
# Adjusting the following parameters is optional:
beegfs_ha_alert_timestamp_format: "%Y-%m-%d %H:%M:%S.%N" #%H:%M:%S.%N
beegfs_ha_alert_verbosity: 3
#  1) high-level node activity
#  3) high-level node activity + fencing action information + resources
(filter on X-monitor)
#  5) high-level node activity + fencing action information + resources
```

儘管看似冗餘、但當您將BeeGFS檔案系統擴充至單一HA叢集以外的位置時、「beegfs_ha_mgmtd_浮 點_ip'是很重要的。部署後續HA叢集時、不需要額外的BeeGFS管理服務、並指向第一個叢集所提供的管理服務。

3. 設定隔離代理程式。（如需詳細資訊、請參閱 "在Red Hat High Availability叢集中設定隔離功能"。）下列輸出顯示設定一般隔離代理程式的範例。請選擇下列其中一個選項。

   在此步驟中、請注意：

- 預設會啟用隔離功能、但您需要設定隔離_agent_。
- 在「PCM1_host_map」或「PCM1_host_list」中指定的「<主機名稱>」必須對應至「Ansible」清單中的主機名稱。
- 不支援在沒有隔離的情況下執行BeeGFS叢集、尤其是在正式作業中。這主要是為了確保BeeGFS服務（包括區塊裝置等任何資源相依性）因發生問題而容錯移轉、不會有多個節點同時存取的風險、進而導致檔案系統毀損或其他不良或非預期的行為。如果必須停用隔離功能、請參閱BeeGFS HA角色使用入門指南中的一般附註、並在「ha_cluster_crm_config_options[stonith啟用的]中、將「beegfs_ha_cluster_crm_config_options[stonith啟用的]」設為「假」。
- 有多個節點層級的隔離裝置可供使用、BeeGFS HA角色可設定Red Hat HA套件儲存庫中可用的任何隔離代理程式。如果可能、請使用透過不斷電系統（UPS）或機架電力分配單元（rPDU）運作的隔離代理程式、 由於某些隔離代理程式（例如基板管理控制器（BMC）或伺服器內建的其他熄燈裝置）、在某些故障情況下可能無法回應Fence要求。

```
### Fencing configuration:
# OPTION 1: To enable fencing using APC Power Distribution Units
(PDUs):
beegfs_ha_fencing_agents:
 fence_apc:
   - ipaddr: <PDU_IP_ADDRESS>
     login: "{{ apc_username }}" # Parameter for APC PDU username in
the passwords file.
     passwd: "{{ apc_password }}" # Parameter for APC PDU password in
the passwords file.
     pcmk_host_map:
"<HOSTNAME>:<PDU_PORT>,<PDU_PORT>;<HOSTNAME>:<PDU_PORT>,<PDU_PORT>"
# OPTION 2: To enable fencing using the Redfish APIs provided by the
Lenovo XCC (and other BMCs):
redfish: &redfish
  username: "{{ bmc_username }}" # Parameter for XCC/BMC username in
the passwords file.
  password: "{{ bmc_password }}" # Parameter for XCC/BMC password in
the passwords file.
    ssl_insecure: 1 # If a valid SSL certificate is not available
specify “1”.
beegfs_ha_fencing_agents:
  fence_redfish:
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish
# For details on configuring other fencing agents see
https://access.redhat.com/documentation/en-
us/red_hat_enterprise_linux/9/html/configuring_and_managing_high_avai
lability_clusters/assembly_configuring-fencing-configuring-and-
managing-high-availability-clusters.
```

4. 在Linux作業系統中啟用建議的效能調校。

   雖然許多使用者認為效能參數的預設設定通常運作良好、但您可以選擇變更特定工作負載的預設設定。因此、這些建議會包含在BeeGFS角色中、但預設不會啟用、以確保使用者知道套用至其檔案系統的調校。

   若要啟用效能調校、請指定：

```
### Performance Configuration:
beegfs_ha_enable_performance_tuning: True
```

5. （選用）您可以視需要調整Linux作業系統中的效能調校參數。

   如需您可以調整的可用調校參數完整清單，請參閱中 BeeGFS HA 角色的效能調校預設值一節 "E系列BeeGFS GitHub網站"。 此檔案中叢集中的所有節點或個別節點的檔案都可以覆寫預設值 `host_vars` 。

6. 若要在區塊和檔案節點之間提供完整的 200GB/HDR 連線能力、請使用 NVIDIA 開放式 Fabric 企業配送（ MLNX_OFED ）中的開放式子網路管理員（ OpenSM ）套件。所列的 MLNx_OFED 版本 "檔案節點需求" 隨附於建議的 OpenSM 套件。雖然支援使用 Ansible 進行部署、但您必須先在所有檔案節點上安裝 MLNX_OFED 驅動程式。

   a. 在「group vars/ha_cluster・yml」（視需要調整套件）中填入下列參數：

   ```
   ### OpenSM package and configuration information
   eseries_ib_opensm_options:
     virt_enabled: "2"
     virt_max_ports_in_process: "0"
   ```

7. 設定「udev"規則、確保邏輯InfiniBand連接埠識別碼與基礎PCIe裝置之間的對應一致。

   「udev"規則必須是每個作為BeeGFS檔案節點之伺服器平台的PCIe拓撲所特有的規則。

   驗證的檔案節點請使用下列值：

   ```
   ### Ensure Consistent Logical IB Port Numbering
   # OPTION 1: Lenovo SR665 V3 PCIe address-to-logical IB port mapping:
   eseries_ipoib_udev_rules:
     "0000:01:00.0": i1a
     "0000:01:00.1": i1b
     "0000:41:00.0": i2a
     "0000:41:00.1": i2b
     "0000:81:00.0": i3a
     "0000:81:00.1": i3b
     "0000:a1:00.0": i4a
     "0000:a1:00.1": i4b

   # OPTION 2: Lenovo SR665 PCIe address-to-logical IB port mapping:
   eseries_ipoib_udev_rules:
     "0000:41:00.0": i1a
     "0000:41:00.1": i1b
     "0000:01:00.0": i2a
     "0000:01:00.1": i2b
     "0000:a1:00.0": i3a
     "0000:a1:00.1": i3b
     "0000:81:00.0": i4a
     "0000:81:00.1": i4b
   ```

8. （選用）更新中繼資料目標選取演算法。

```
beegfs_ha_beegfs_meta_conf_ha_group_options:
  tuneTargetChooser: randomrobin
```

> 在驗證測試中、「隨機配置資源」通常用於確保測試檔案在效能基準測試期間平均分散到所有BeeGFS儲存目標（如需基準測試的詳細資訊、請參閱BeeGFS網站 "基準測試BeeGFS系統"）。實際使用時、可能會導致編號較低的目標填滿速度比編號較高的目標更快。省略「Randomrounds」、只要使用預設的「Randomized」（隨機）值、就能提供良好的效能、同時仍能使用所有可用的目標。

步驟**5**：定義通用區塊節點的組態

區塊節點的共用組態是在稱為「Eseria_storage系統」的群組中定義。本節中的步驟會建置應包含在「group _vars/ Eseries _storage系統.yml」檔案中的組態。

步驟

1. 設定「Ansible connection to local（可連線至本機）」、提供系統密碼、並指定是否應驗證SSL憑證。（通常情況下、Ansible會使用SSH連線至託管主機、但在使用NetApp E系列儲存系統做為區塊節點的情況下、模組會使用REST API進行通訊。） 在檔案頂端新增下列項目：

```
### eseries_storage_systems Ansible group inventory file.
# Place all default/common variables for NetApp E-Series Storage Systems
here:
ansible_connection: local
eseries_system_password: {{ eseries_password }} # Parameter for E-Series
storage array password in the passwords file.
eseries_validate_certs: false
```

2. 若要確保最佳效能、請在中安裝區塊節點所列的版本 "技術需求"。

   請從下載對應的檔案 "NetApp支援網站"。您可以手動升級、或是將它們納入Ansible控制節點的「套件/」目錄、然後在「Eserie_storage儲存系統.yml」中填入下列參數、以使用Ansible進行升級：

```
# Firmware, NVSRAM, and Drive Firmware (modify the filenames as needed):
eseries_firmware_firmware: "packages/RCB_11.80GA_6000_64cc0ee3.dlp"
eseries_firmware_nvsram: "packages/N6000-880834-D08.dlp"
```

3. 從下載並安裝適用於區塊節點中安裝之磁碟機的最新磁碟機韌體 "NetApp支援網站"。您可以手動升級它們、或將它們納入 `packages/` Ansible 控制節點的目錄、然後在中填入下列參數 `eseries_storage_systems.yml` 、以使用 Ansible 進行升級：

```
eseries_drive_firmware_firmware_list:
  - "packages/<FILENAME>.dlp"
eseries_drive_firmware_upgrade_drives_online: true
```

> 將「Eseria_drive_韌 體_grade_drives_online」設定為「假」會加速升級、但必須等到部署BeeGFS之後才能執行。這是因為該設定需要在升級前停止所有磁碟機的I/O、以避免應用程式錯誤。雖然在設定磁碟區之前執行線上磁碟機韌體升級仍很快、但我們建議您將此值設為「true」、以避免日後發生問題。

4. 若要最佳化效能、請對全域組態進行下列變更：

```
# Global Configuration Defaults
eseries_system_cache_block_size: 32768
eseries_system_cache_flush_threshold: 80
eseries_system_default_host_type: linux dm-mp
eseries_system_autoload_balance: disabled
eseries_system_host_connectivity_reporting: disabled
eseries_system_controller_shelf_id: 99 # Required.
```

5. 若要確保最佳的Volume資源配置和行為、請指定下列參數：

```
# Storage Provisioning Defaults
eseries_volume_size_unit: pct
eseries_volume_read_cache_enable: true
eseries_volume_read_ahead_enable: false
eseries_volume_write_cache_enable: true
eseries_volume_write_cache_mirror_enable: true
eseries_volume_cache_without_batteries: false
eseries_storage_pool_usable_drives:
"99:0,99:23,99:1,99:22,99:2,99:21,99:3,99:20,99:4,99:19,99:5,99:18,99:6,
99:17,99:7,99:16,99:8,99:15,99:9,99:14,99:10,99:13,99:11,99:12"
```

> 針對「Eseria_storage資源池可用磁碟機」指定的值、是NetApp EF600區塊節點的專屬值、可控制磁碟機指派給新Volume群組的順序。此順序可確保每個群組的I/O平均分散於後端磁碟機通道。

定義**BeeGFS**建置區塊的**Ansible**庫存

定義一般的Ansible庫存結構之後、請定義BeeGFS檔案系統中每個建置區塊的組態。

這些部署說明示範如何部署由基礎建置區塊（包括管理、中繼資料和儲存服務）所組成的檔案系統、第二個含有中繼資料和儲存服務的建置區塊、以及第三個純儲存建置區塊。

這些步驟旨在顯示完整的典型組態設定檔、您可以使用這些設定檔來設定NetApp BeeGFS建置區塊、以符合整個BeeGFS檔案系統的需求。

> 在本節及後續章節中、視需要進行調整、以建立代表您要部署之BeeGFS檔案系統的詳細目錄。尤其是使用代表每個區塊或檔案節點的Ansible主機名稱、以及儲存網路所需的IP定址方案、以確保其可擴充至BeeGFS檔案節點和用戶端的數量。

步驟1：建立Ansible庫存檔案

步驟

1. 建立新的「inventory.yml」檔案、然後插入下列參數、視需要將主機替換為「Eseria_storage系統」、以代表部署中的區塊節點。名稱應與「host_vars/<fileName（主機名稱）>.yml」所使用的名稱相對應。

```
# BeeGFS HA (High Availability) cluster inventory.
all:
  children:
    # Ansible group representing all block nodes:
    eseries_storage_systems:
      hosts:
        netapp_01:
        netapp_02:
        netapp_03:
        netapp_04:
        netapp_05:
        netapp_06:
    # Ansible group representing all file nodes:
    ha_cluster:
      children:
```

在後續章節中、您將在「ha_cluster」下建立其他可執行群組、以代表您要在叢集中執行的BeeGFS服務。

步驟2：設定管理、中繼資料和儲存建置區塊的庫存

叢集或基礎建置區塊中的第一個建置區塊必須包含BeeGFS管理服務、以及中繼資料和儲存服務：

步驟

1. 在「inventory.yml」中、在「ha_cluster：子項目」下填入下列參數：

```
      # beegfs_01/beegfs_02 HA Pair (mgmt/meta/storage building block):
        mgmt:
          hosts:
            beegfs_01:
            beegfs_02:
        meta_01:
          hosts:
            beegfs_01:
```

```
    beegfs_02:
stor_01:
  hosts:
    beegfs_01:
    beegfs_02:
meta_02:
  hosts:
    beegfs_01:
    beegfs_02:
stor_02:
  hosts:
    beegfs_01:
    beegfs_02:
meta_03:
  hosts:
    beegfs_01:
    beegfs_02:
stor_03:
  hosts:
    beegfs_01:
    beegfs_02:
meta_04:
  hosts:
    beegfs_01:
    beegfs_02:
stor_04:
  hosts:
    beegfs_01:
    beegfs_02:
meta_05:
  hosts:
    beegfs_02:
    beegfs_01:
stor_05:
  hosts:
    beegfs_02:
    beegfs_01:
meta_06:
  hosts:
    beegfs_02:
    beegfs_01:
stor_06:
  hosts:
    beegfs_02:
    beegfs_01:
meta_07:
```

```
          hosts:
            beegfs_02:
            beegfs_01:
        stor_07:
          hosts:
            beegfs_02:
            beegfs_01:
        meta_08:
          hosts:
            beegfs_02:
            beegfs_01:
        stor_08:
          hosts:
            beegfs_02:
            beegfs_01:
```

2. 建立「group vars/mgmt.ml」檔案、並包含下列項目：

```
# mgmt - BeeGFS HA Management Resource Group
# OPTIONAL: Override default BeeGFS management configuration:
# beegfs_ha_beegfs_mgmtd_conf_resource_group_options:
#  <beegfs-mgmt.conf:key>:<beegfs-mgmt.conf:value>
floating_ips:
  - i1b: 100.127.101.0/16
  - i2b: 100.127.102.0/16
beegfs_service: management
beegfs_targets:
  netapp_01:
    eseries_storage_pool_configuration:
      - name: beegfs_m1_m2_m5_m6
        raid_level: raid1
        criteria_drive_count: 4
        common_volume_configuration:
          segment_size_kb:  128
        volumes:
          - size: 1
            owning_controller: A
```

3. 在「Group_vars/」下、使用下列範本建立資源群組「meta_01」到「meta_08」的檔案、然後填寫下表中每個服務的預留位置值：

```
# meta_0X - BeeGFS HA Metadata Resource Group
beegfs_ha_beegfs_meta_conf_resource_group_options:
  connMetaPortTCP: <PORT>
  connMetaPortUDP: <PORT>
  tuneBindToNumaZone: <NUMA ZONE>
floating_ips:
  - <PREFERRED PORT:IP/SUBNET> # Example: i1b:192.168.120.1/16
  - <SECONDARY PORT:IP/SUBNET>
beegfs_service: metadata
beegfs_targets:
  <BLOCK NODE>:
    eseries_storage_pool_configuration:
      - name: <STORAGE POOL>
        raid_level: raid1
        criteria_drive_count: 4
        common_volume_configuration:
          segment_size_kb:  128
        volumes:
          - size: 21.25 # SEE NOTE BELOW!
            owning_controller: <OWNING CONTROLLER>
```

磁碟區大小是以整體儲存資源池（也稱為Volume群組）的百分比來指定。NetApp強烈建議您在每個資源池中保留一些可用容量、以便有空間進行SSD過度資源配置（如需詳細資訊、請參閱 "NetApp EF600陣列簡介"）。儲存資源池「beegfs_m1_m2_m5_m6'」也會將1%的資源池容量配置給管理服務。因此、對於儲存資源池中的中繼資料磁碟區、當使用1.92TB或3.844TB磁碟機時、請將此值設為「21.25」；如果使用7.65TB磁碟機、請將此值設為「22.25」；如果使用15.3TB磁碟機、請將此值設為「23.75」。

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
meta_01.yml	8015	i1b ： 100.127.101.1/16 i2b ： 100.127.102.1/16	0	netapp_01	beegfs_m1_m2_m5_m6.	答
meta_02.yml	8025	i2b ： 100.127.102.2/16 i1b ： 100.127.101.2/16	0	netapp_01	beegfs_m1_m2_m5_m6.	b
meta_03.yml	8035	i3b ： 100.127.101.3/16 i4b ： 100.127.102.3/16	1.	netapp_02	Beegfs_m3_m4_m7_M8	答

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
meta_04.yml	8045	i4b ： 100.127.102.4/16 i3b ： 100.127.101.4/16	1.	netapp_02	Beegfs_m3_m4_m7_M8	b
meta_05.yml	8055	i1b ： 100.127.101.5/16 i2b ： 100.127.102.5/16	0	netapp_01	beegfs_m1_m2_m5_m6.	答
meta_06.yml	8065	i2b ： 100.127.102.6/16 i1b ： 100.127.101.6/16	0	netapp_01	beegfs_m1_m2_m5_m6.	b
meta_07.yml	8075	i3b ： 100.127.101.7/16 i4b ： 100.127.102.7/16	1.	netapp_02	Beegfs_m3_m4_m7_M8	答
meta_08.yml	8085	i4b ： 100.127.102.8/16 i3b ： 100.127.101.8/16	1.	netapp_02	Beegfs_m3_m4_m7_M8	b

4. 在「Group_vars/」下、使用下列範本建立資源群組「shor_01」到「shor_08」的檔案、然後填入每個服務的預留位置值、以參照範例：

```
# stor_0X - BeeGFS HA Storage Resource
Groupbeegfs_ha_beegfs_storage_conf_resource_group_options:
  connStoragePortTCP: <PORT>
  connStoragePortUDP: <PORT>
  tuneBindToNumaZone: <NUMA ZONE>
floating_ips:
  - <PREFERRED PORT:IP/SUBNET>
  - <SECONDARY PORT:IP/SUBNET>
beegfs_service: storage
beegfs_targets:
  <BLOCK NODE>:
    eseries_storage_pool_configuration:
      - name: <STORAGE POOL>
        raid_level: raid6
        criteria_drive_count: 10
        common_volume_configuration:
          segment_size_kb: 512        volumes:
          - size: 21.50 # See note below!             owning_controller:
<OWNING CONTROLLER>
          - size: 21.50            owning_controller: <OWNING
CONTROLLER>
```

如需正確使用尺寸、請參閱 "建議的儲存資源池過度資源配置百分比"。

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
STOR_01.yml	8013	i1b ： 100.127.103. 1/16 i2b ： 100.127.104. 1/16	0	netapp_01	beegfs_s1_s2	答
STOR_02.yml	8023	i2b ： 100.127.104. 2/16 i1b ： 100.127.103. 2/16	0	netapp_01	beegfs_s1_s2	b
STOR_03.yml	8033	i3b ： 100.127.103. 3/16 i4b ： 100.127.104. 3/16	1.	netapp_02	beegfs_s2_s4	答
STOR_04.yml	8043	i4b ： 100.127.104. 4/16 i3b ： 100.127.103. 4/16	1.	netapp_02	beegfs_s2_s4	b

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
STOR_05.yml	8053	i1b ： 100.127.103.5/16 i2b ： 100.127.104.5/16	0	netapp_01	Beegfs_S1_S6	答
STOR_06.yml	8063	i2b ： 100.127.104.6/16 i1b ： 100.127.103.6/16	0	netapp_01	Beegfs_S1_S6	b
STOR_07.yml	8073	i3b ： 100.127.103.7/16 i4b ： 100.127.104.7/16	1.	netapp_02	Beegfs_S7_S8	答
STOR_08.yml	8083	i4b ： 100.127.104.8/16 i3b ： 100.127.103.8/16	1.	netapp_02	Beegfs_S7_S8	b

步驟**3**：設定中繼資料**+**儲存建置區塊的詳細目錄

這些步驟說明如何設定BeeGFS中繼資料+儲存建置區塊的可執行庫存。

步驟

1. 在「inventory.yml」中、在現有組態下填入下列參數：

```
        meta_09:
          hosts:
            beegfs_03:
            beegfs_04:
        stor_09:
          hosts:
            beegfs_03:
            beegfs_04:
        meta_10:
          hosts:
            beegfs_03:
            beegfs_04:
        stor_10:
          hosts:
            beegfs_03:
            beegfs_04:
        meta_11:
```

```
      hosts:
        beegfs_03:
        beegfs_04:
    stor_11:
      hosts:
        beegfs_03:
        beegfs_04:
    meta_12:
      hosts:
        beegfs_03:
        beegfs_04:
    stor_12:
      hosts:
        beegfs_03:
        beegfs_04:
    meta_13:
      hosts:
        beegfs_04:
        beegfs_03:
    stor_13:
      hosts:
        beegfs_04:
        beegfs_03:
    meta_14:
      hosts:
        beegfs_04:
        beegfs_03:
    stor_14:
      hosts:
        beegfs_04:
        beegfs_03:
    meta_15:
      hosts:
        beegfs_04:
        beegfs_03:
    stor_15:
      hosts:
        beegfs_04:
        beegfs_03:
    meta_16:
      hosts:
        beegfs_04:
        beegfs_03:
    stor_16:
      hosts:
        beegfs_04:
```

```
            beegfs_03:
```

2. 在「Group_vars/」下、使用下列範本建立資源群組「meta_09」到「meta_16」的檔案、然後填入每個服務的預留位置值、以參照範例：

```
# meta_0X - BeeGFS HA Metadata Resource Group
beegfs_ha_beegfs_meta_conf_resource_group_options:
  connMetaPortTCP: <PORT>
  connMetaPortUDP: <PORT>
  tuneBindToNumaZone: <NUMA ZONE>
floating_ips:
  - <PREFERRED PORT:IP/SUBNET>
  - <SECONDARY PORT:IP/SUBNET>
beegfs_service: metadata
beegfs_targets:
  <BLOCK NODE>:
    eseries_storage_pool_configuration:
      - name: <STORAGE POOL>
        raid_level: raid1
        criteria_drive_count: 4
        common_volume_configuration:
          segment_size_kb: 128
        volumes:
          - size: 21.5 # SEE NOTE BELOW!
            owning_controller: <OWNING CONTROLLER>
```

如需正確使用尺寸、請參閱 "建議的儲存資源池過度資源配置百分比"。

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
meta_09.yml	8015	i1b ： 100.127.101.9/16 i2b ： 100.127.102.9/16	0	netapp_03	Beegfs_m9_m10_M13_M14	答
meta_10.yml	8025	i2b:100.127.102.10/16 i1b:100.127.101.10/16	0	netapp_03	Beegfs_m9_m10_M13_M14	b
meta_11.ml	8035	i3b ： 100.127.101.11/16 i4b ： 100.127.102.11/16	1.	netapp_04	Beegfs_M11_M12_M15_M16	答

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
meta_12.ml	8045	i4b ： 100.127.102.12/16 i3b ： 100.127.101.12/16	1.	netapp_04	Beegfs_M11_M12_M15_M16	b
meta_13.yml	8055	i1b ： 100 、 127.101.3/16 i2b ： 100 、 127.102.3/16	0	netapp_03	Beegfs_m9_m10_M13_M14	答
meta_14.yml	8065	i2b:100.127.102.14/16 i1b:100.127.101.14/16	0	netapp_03	Beegfs_m9_m10_M13_M14	b
meta_15.yml	8075	i3b ： 100.127.101.15/16 i4b ： 100.127.102.15/16	1.	netapp_04	Beegfs_M11_M12_M15_M16	答
meta_16.myl	8085	i4b ： 100.127.102.16/16 i3b ： 100.127.101.16/16	1.	netapp_04	Beegfs_M11_M12_M15_M16	b

3. 在「Group_vars/」下、使用下列範本建立資源群組「shor_09」到「shor_16」的檔案、然後填入每個服務的預留位置值、以參照範例：

```
# stor_0X - BeeGFS HA Storage Resource Group
beegfs_ha_beegfs_storage_conf_resource_group_options:
  connStoragePortTCP: <PORT>
  connStoragePortUDP: <PORT>
  tuneBindToNumaZone: <NUMA ZONE>
floating_ips:
  - <PREFERRED PORT:IP/SUBNET>
  - <SECONDARY PORT:IP/SUBNET>
beegfs_service: storage
beegfs_targets:
  <BLOCK NODE>:
    eseries_storage_pool_configuration:
      - name: <STORAGE POOL>
        raid_level: raid6
        criteria_drive_count: 10
        common_volume_configuration:
          segment_size_kb: 512        volumes:
          - size: 21.50 # See note below!
            owning_controller: <OWNING CONTROLLER>
          - size: 21.50            owning_controller: <OWNING
CONTROLLER>
```

要了解正確的尺寸，請參閱"建議的儲存資源池過度資源配置百分比" ..

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
STOR_09.yml	8013	i1b ： 100.127.103.9/16 i2b ： 100.127.104.9/16	0	netapp_03	beegfs_s9_s10	答
STOR_10.yml	8023	i2b:100.127.104.10/16 i1b:100.127.103.10/16	0	netapp_03	beegfs_s9_s10	b
STOR_11.yml	8033	i3b ： 100.127.103.11/16 i4b ： 100.127.104.11/16	1.	netapp_04	Beegfs_S11_s12.	答
Stor_12.ml	8043	i4b ： 100.127.104.12/16 i3b ： 100.127.103.12/16	1.	netapp_04	Beegfs_S11_s12.	b

檔案名稱	連接埠	浮動**IP**	**NUMA**區域	區塊節點	儲存資源池	擁有控制器
STOR_13.yml	8053	i1b ： 100.127.103.13/16 i2b ： 100.127.104.13/16	0	netapp_03	beegfs_s13_s14	答
STOR_14.yml	8063	i2b:100.127.104.14/16 i1b:100.127.103.14/16	0	netapp_03	beegfs_s13_s14	b
STOR_15.yml	8073	i3b ： 100.127.103.15/16 i4b ： 100.127.104.15/16	1.	netapp_04	Beegfs_S15_S16	答
STOR_16.yml	8083	i4b ： 100.127.104.16/16 i3b ： 100.127.103.16/16	1.	netapp_04	Beegfs_S15_S16	b

步驟**4**：設定僅儲存建置區塊的庫存

這些步驟說明如何設定BeeGFS純儲存區塊的可執行庫存。設定中繼資料+儲存設備的組態與純儲存設備建置區塊之間的主要差異、在於所有中繼資料資源群組都不存在、而且每個儲存資源池的「Criteria _DRIVE _count」也會從10變更為12。

步驟

1. 在「inventory.yml」中、在現有組態下填入下列參數：

```
# beegfs_05/beegfs_06 HA Pair (storage only building block):
  stor_17:
    hosts:
      beegfs_05:
      beegfs_06:
  stor_18:
    hosts:
      beegfs_05:
      beegfs_06:
  stor_19:
    hosts:
      beegfs_05:
      beegfs_06:
  stor_20:
    hosts:
      beegfs_05:
      beegfs_06:
  stor_21:
    hosts:
      beegfs_06:
      beegfs_05:
  stor_22:
    hosts:
      beegfs_06:
      beegfs_05:
  stor_23:
    hosts:
      beegfs_06:
      beegfs_05:
  stor_24:
    hosts:
      beegfs_06:
      beegfs_05:
```

2. 在「Group_vars/」下、使用下列範本建立資源群組「shor_17」到「shor_24」的檔案、然後填寫每個服務的預留位置值、以參照範例：

```
# stor_0X - BeeGFS HA Storage Resource Group
beegfs_ha_beegfs_storage_conf_resource_group_options:
  connStoragePortTCP: <PORT>
  connStoragePortUDP: <PORT>
  tuneBindToNumaZone: <NUMA ZONE>
floating_ips:
  - <PREFERRED PORT:IP/SUBNET>
  - <SECONDARY PORT:IP/SUBNET>
beegfs_service: storage
beegfs_targets:
  <BLOCK NODE>:
    eseries_storage_pool_configuration:
      - name: <STORAGE POOL>
        raid_level: raid6
        criteria_drive_count: 12
        common_volume_configuration:
          segment_size_kb: 512
        volumes:
          - size: 21.50 # See note below!
            owning_controller: <OWNING CONTROLLER>
          - size: 21.50
            owning_controller: <OWNING CONTROLLER>
```

要了解正確的尺寸，請參閱"建議的儲存資源池過度資源配置百分比"。

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
STOR_17.yml	8013	i1b ： 100.127.103.17/16 i2b ： 100.127.104.17/16	0	netapp_05	Beegfs_S17_s18	答
STOR_18.yml	8023	i2b:100.127.104.18/16 i1b:100.127.103.18/16	0	netapp_05	Beegfs_S17_s18	b
STOR_19.yml	8033	i3b ： 100.127.103.19/16 i4b ： 100.127.104.19/16	1.	netapp_06	Beegfs_s19_S20	答

檔案名稱	連接埠	浮動IP	NUMA區域	區塊節點	儲存資源池	擁有控制器
Stor_20.ml	8043	i4b ： 100 、 127.104.20/16 i3b ： 100 、 127.103.20/16	1.	netapp_06	Beegfs_s19_S20	b
STOR_21.yml	8053	i1b ： 100.127.103.21/16 i2b ： 100.127.104.21/16	0	netapp_05	Beegfs_S21_S22	答
STOR_22.yml	8063	i2b:100.127.104.22/16 i1b:100.127.103.22/16	0	netapp_05	Beegfs_S21_S22	b
STOR_23.yml	8073	i3b ： 100.127.103.23/16 i4b ： 100.127.104.23/16	1.	netapp_06	beegfs_S23_s24	答
STOR_24.yml	8083	i4b ： 100.127.104.24/16 i3b ： 100.127.103.24/16	1.	netapp_06	beegfs_S23_s24	b

部署**BeeGFS**

部署及管理組態時、需要執行一或多個包含執行必要工作的教戰手冊、並將整體系統移至所需狀態。

雖然所有工作都可納入單一教戰手冊中、但對於複雜的系統而言、這種做法很快就變得難以管理。Ansible可讓您建立及發佈角色、以封裝可重複使用的教戰手冊和相關內容（例如：預設變數、工作和處理常式）。如需詳細資訊、請參閱的「Ansible」文件 "角色"。

角色通常會在包含相關角色和模組的可Ansible集合中散佈。因此、這些教戰手冊主要是匯入分散在各種NetApp E系列Ansible系列收藏中的多個角色。

目前、部署BeeGFS至少需要兩個建置區塊（四個檔案節點）、除非將個別的仲裁裝置設定為連線斷路器、以減輕在使用雙節點叢集建立仲裁時發生的任何問題。

步驟

1. 建立新的「playbook、yml」檔案、其中包括：

```
# BeeGFS HA (High Availability) cluster playbook.
- hosts: eseries_storage_systems
```

```
  gather_facts: false
  collections:
    - netapp_eseries.santricity
  tasks:
    - name: Configure NetApp E-Series block nodes.
      import_role:
        name: nar_santricity_management
- hosts: all
  any_errors_fatal: true
  gather_facts: false
  collections:
    - netapp_eseries.beegfs
  pre_tasks:
    - name: Ensure a supported version of Python is available on all
file nodes.
      block:
        - name: Check if python is installed.
          failed_when: false
          changed_when: false
          raw: python --version
          register: python_version
        - name: Check if python3 is installed.
          raw: python3 --version
          failed_when: false
          changed_when: false
          register: python3_version
          when: 'python_version["rc"] != 0 or (python_version["stdout"]
| regex_replace("Python ", "")) is not version("3.0", ">=")'
        - name: Install python3 if needed.
          raw: |
            id=$(grep "^ID=" /etc/*release* | cut -d= -f 2 | tr -d '"')
            case $id in
              ubuntu) sudo apt install python3 ;;
              rhel|centos) sudo yum -y install python3 ;;
              sles) sudo zypper install python3 ;;
            esac
          args:
            executable: /bin/bash
          register: python3_install
          when: python_version['rc'] != 0 and python3_version['rc'] != 0
          become: true
        - name: Create a symbolic link to python from python3.
          raw: ln -s /usr/bin/python3 /usr/bin/python
          become: true
          when: python_version['rc'] != 0
      when: inventory_hostname not in
```

```
groups[beegfs_ha_ansible_storage_group]
    - name: Verify any provided tags are supported.
      fail:
        msg: "{{ item }} tag is not a supported BeeGFS HA tag. Rerun
your playbook command with --list-tags to see all valid playbook tags."
      when: 'item not in ["all", "storage", "beegfs_ha",
"beegfs_ha_package", "beegfs_ha_configure",
"beegfs_ha_configure_resource", "beegfs_ha_performance_tuning",
"beegfs_ha_backup", "beegfs_ha_client"]'
      loop: "{{ ansible_run_tags }}"
  tasks:
    - name: Verify before proceeding.
      pause:
        prompt: "Are you ready to proceed with running the BeeGFS HA
role? Depending on the size of the deployment and network performance
between the Ansible control node and BeeGFS file and block nodes this
can take awhile (10+ minutes) to complete."
    - name: Verify the BeeGFS HA cluster is properly deployed.
      ansible.builtin.import_role:
        name: netapp_eseries.beegfs.beegfs_ha_7_4
```

本方針執行幾項「pre_tesss」、以驗證檔案節點上是否安裝Python 3、並檢查所提供的Ansible標記是否受到支援。

2. 當您準備部署BeeGFS時、請將「Ansible Playbook」命令與庫存和方針檔案搭配使用。

   部署作業會執行所有的「pre_tessment」、然後在繼續實際部署BeeGFS之前提示使用者確認。

   執行下列命令、視需要調整貨叉數量（請參閱以下附註）：

```
ansible-playbook -i inventory.yml playbook.yml --forks 20
```

特別是對於較大型的部署、`forks`建議使用參數覆寫預設的叉具（5）、以增加 Ansible 平行設定的主機數量。（如需詳細資訊 "控制方針執行"、請參閱。）最大值設定取決於 Ansible 控制節點上可用的處理能力。上述20個範例是在具有4個CPU（Intel（R）Xeon（R）Gold 6146 CPU @ 3.20GHz）的虛擬Ansible控制節點上執行。

視部署規模和Ansible控制節點與BeeGFS檔案和區塊節點之間的網路效能而定、部署時間可能會有所不同。

設定**BeeGFS**用戶端

您必須在需要存取BeeGFS檔案系統的任何主機（例如運算或GPU節點）上安裝及設定BeeGFS用戶端。在這項工作中、您可以使用Ansible和BeeGFS集合。

步驟

1. 如有需要、請從Ansible控制節點設定無密碼SSH、並將其設定為BeeGFS用戶端的每個主機：

   「sh-copy -id <user>@<hostname_or_ip>'」

2. 在「host_vars/」下、為每個BeeGFS用戶端建立一個名為「.yml」的檔案、其中包含下列內容、並在預留位置文字中填入適合您環境的正確資訊：

```
# BeeGFS Client
ansible_host: <MANAGEMENT_IP>
# OPTIONAL: If you want to use the NetApp E-Series Host Collection's
IPoIB role to configure InfiniBand interfaces for clients to connect to
BeeGFS file systems:
eseries_ipoib_interfaces:
  - name: <INTERFACE>
    address: <IP>/<SUBNET_MASK> # Example: 100.127.1.1/16
  - name: <INTERFACE>
    address: <IP>/<SUBNET_MASK>
```

   如果使用兩個子網路定址方案進行部署、則必須在每個用戶端上設定兩個 InfiniBand 介面、每個都在兩個儲存 IPoIB 子網路中設定一個。如果針對此處列出的每項 BeeGFS 服務使用範例子網路和建議範圍、則用戶端應在「至」範圍內設定一個介面、而在「至」中設定另一個介面 `100.127.1.0` `100.127.99.255` `100.128.1.0` `100.128.99.255`。

3. 建立新檔案「client_inventory.yml」、然後在頂端填入下列參數：

```
# BeeGFS client inventory.
all:
  vars:
    ansible_ssh_user: <USER> # This is the user Ansible should use to
connect to each client.
    ansible_become_password: <PASSWORD> # This is the password Ansible
will use for privilege escalation, and requires the ansible_ssh_user be
root, or have sudo privileges.
The defaults set by the BeeGFS HA role are based on the testing
performed as part of this NetApp Verified Architecture and differ from
the typical BeeGFS client defaults.
```

   請勿以純文字儲存密碼。請改用Ansible Vault（請參閱的「Ansible」文件） "使用Ansible Vault加密內容"）或是在執行該教戰手冊時使用「Ask（隨叫隨到）」選項。

4. 在「client_inventory.yml」檔案中、在「beegfs_clients」群組中列出所有應設定為BeeGFS用戶端的主機、然後指定建置BeeGFS用戶端核心模組所需的任何其他組態。

```
  children:
    # Ansible group representing all BeeGFS clients:
    beegfs_clients:
      hosts:
        beegfs_01:
        beegfs_02:
        beegfs_03:
        beegfs_04:
        beegfs_05:
        beegfs_06:
        beegfs_07:
        beegfs_08:
        beegfs_09:
        beegfs_10:
      vars:
        # OPTION 1: If you're using the NVIDIA OFED drivers and they are
already installed:
        eseries_ib_skip: True # Skip installing inbox drivers when using
the IPoIB role.
        beegfs_client_ofed_enable: True
        beegfs_client_ofed_include_path:
"/usr/src/ofa_kernel/default/include"
        # OPTION 2: If you're using inbox IB/RDMA drivers and they are
already installed:
        eseries_ib_skip: True # Skip installing inbox drivers when using
the IPoIB role.
        # OPTION 3: If you want to use inbox IB/RDMA drivers and need
them installed/configured.
        eseries_ib_skip: False # Default value.
        beegfs_client_ofed_enable: False # Default value.
```

使用 NVIDIA OFED 驅動程式時、請 `beegfs_client_ofed_include_path` 務必針對您的 Linux 安裝指向正確的「標頭包含路徑」。如需詳細資訊，請參閱的 BeeGFS 文件 "RDMA 支援"。

5. 在「client_inventory.yml」檔案中、列出您要掛載在任何先前定義「vars」底部的BeeGFS檔案系統。

```
        beegfs_client_mounts:
          - sysMgmtdHost: 100.127.101.0 # Primary IP of the BeeGFS
management service.
            mount_point: /mnt/beegfs    # Path to mount BeeGFS on the
client.
            connInterfaces:
              - <INTERFACE> # Example: ibs4f1
              - <INTERFACE>
            beegfs_client_config:
              # Maximum number of simultaneous connections to the same
node.
              connMaxInternodeNum: 128 # BeeGFS Client Default: 12
              # Allocates the number of buffers for transferring IO.
              connRDMABufNum: 36 # BeeGFS Client Default: 70
              # Size of each allocated RDMA buffer
              connRDMABufSize: 65536 # BeeGFS Client Default: 8192
              # Required when using the BeeGFS client with the shared-
disk HA solution.
              # This does require BeeGFS targets be mounted in the
default "sync" mode.
              # See the documentation included with the BeeGFS client
role for full details.
              sysSessionChecksEnabled: false
```

「beegfs_client_config」代表已測試的設定。如需所有選項的完整概觀、請參閱netapp_eseries.beegfs`集合「beegfs_client」角色隨附的文件。這包括有關安裝多個BeeGFS檔案系統或多次安裝同一個BeeGFS檔案系統的詳細資料。

6. 建立新的「client_playbook .yml」檔案、然後填入下列參數：

```
# BeeGFS client playbook.
- hosts: beegfs_clients
  any_errors_fatal: true
  gather_facts: true
  collections:
    - netapp_eseries.beegfs
    - netapp_eseries.host
  tasks:
    - name: Ensure IPoIB is configured
      import_role:
        name: ipoib
    - name: Verify the BeeGFS clients are configured.
      import_role:
        name: beegfs_client
```

如果您已在適當的IPoIB介面上安裝必要的IB/RDMA驅動程式和設定的IP、請省略匯入「NetApp_Eseries.host」集合和「IPoIB」角色。

7. 若要安裝及建置用戶端和Mount BeeGFS、請執行下列命令：

```
ansible-playbook -i client_inventory.yml client_playbook.yml
```

8. 在您將BeeGFS檔案系統置於正式作業環境之前、我們*強烈*建議您登入任何用戶端、然後執行「beegfs-fs-fs-checksfs」、以確保所有節點都可連線、而且不會報告任何問題。

擴充至五個建置區塊以外

您可以設定起搏器和電暈器同步、使其擴充至超過五個建置區塊（10個檔案節點）。不過、較大型的叢集也有缺點、因此心律調整器和電暈器同步最終會強制使用最多32個節點。

NetApp僅針對最多10個節點測試BeeGFS HA叢集、不建議或不支援擴充超過此限制的個別叢集。然而、BeeGFS檔案系統仍需擴充至超過10個節點、而NetApp已在NetApp的BeeGFS解決方案中納入此考量。

透過部署多個HA叢集、其中包含每個檔案系統中的一部分建置區塊、您可以獨立擴充整個BeeGFS檔案系統、使基礎HA叢集機制不受任何建議或硬限制。在此案例中、請執行下列動作：

- 建立代表其他HA叢集的新Ansible庫存、然後省略設定其他管理服務。相反地、將每個額外叢集「ha_cluster.yml」中的「beegfs_ha_mgmtd_浮 點_ip」變數指向第一個BeeGFS管理服務的IP。
- 將其他HA叢集新增至同一個檔案系統時、請確定下列事項：
  - BeeGFS節點ID是唯一的。
  - 與「group vars」下的每個服務對應的檔案名稱、在所有叢集中都是唯一的。
  - BeeGFS用戶端和伺服器IP位址在所有叢集之間都是唯一的。
  - 第一個包含BeeGFS管理服務的HA叢集正在執行、然後才嘗試部署或更新其他叢集。
- 在各自的目錄樹狀結構中分別維護每個HA叢集的庫存。

  嘗試在一個目錄樹狀結構中混合多個叢集的詳細目錄檔案、可能會導致BeeGFS HA角色如何將套用至特定叢集的組態集合在一起時發生問題。

在建立新的HA叢集之前、不需要將每個HA叢集擴充至五個建置區塊。在許多情況下、每個叢集使用較少的建置區塊、更容易管理。一種方法是將每個機架中的建置區塊設定為HA叢集。

建議的儲存資源池過度資源配置百分比

當遵循第二代建置區塊每個儲存池組態的標準四個磁碟區時、請參閱下表。

下表提供每個BeeGFS中繼資料或儲存目標的「Eseria_storage儲存資源池組態」中、作為磁碟區大小的建議百分比：

磁碟機大小	尺寸
1.92TB	18
3.84 TB	21.5
7.68TB	22.5%
15.3TB	24

上述指南不適用於包含管理服務的儲存資源池、此服務應將上述大小減少0.25%、以便將1%的儲存資源池分配給管理資料。

若要瞭解如何判斷這些值、請參閱 "TR-4800:附錄A：瞭解SSD的耐用度和過度資源配置"。

大容量建置區塊

標準BeeGFS解決方案部署指南概述高效能工作負載需求的程序與建議。想要滿足高容量需求的客戶、應觀察此處列出的部署與建議差異。

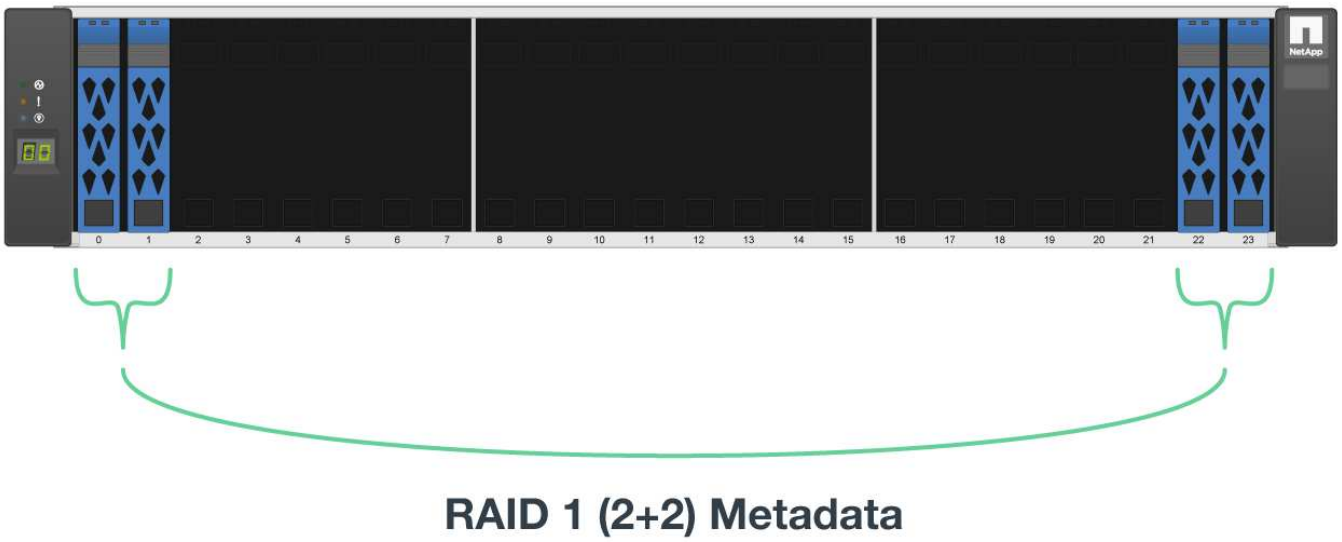
控制器

對於大容量建置區塊、EF600控制器應更換為EF300控制器、每個控制器均安裝Cascade HIC以進行SAS擴充。每個區塊節點在陣列機箱中會有最少數量的NVMe SSD、用於BeeGFS中繼資料儲存設備、並會附加到擴充機櫃、其中會有NL-SAS HDD用於BeeGFS儲存磁碟區。

「檔案節點對區塊」節點組態保持不變。

磁碟機放置

BeeGFS中繼資料儲存設備的每個區塊節點至少需要4個NVMe SSD。這些磁碟機應放置在機箱最外側的插槽中。



擴充托盤

大容量建置區塊的大小可為每個儲存陣列配備1至7個60個磁碟機擴充支架。

如需連接每個擴充托盤的說明、 "請參閱EF300磁碟機櫃纜線"。

使用自訂架構

總覽與需求

使用Ansible部署BeeGFS高可用度叢集時、請將任何NetApp E/EF系列儲存系統當作BeeGFS區塊節點、將x86伺服器當作BeeGFS檔案節點。

本節中使用的術語定義可在"詞彙與概念"頁面中找到。

簡介

雖然"NetApp認證的架構"提供預先定義的參考組態和規模指南、但有些客戶和合作夥伴可能偏好設計更適合特定需求或硬體偏好的自訂架構。在NetApp上選擇BeeGFS的主要優點之一、就是能夠使用Ansible部署BeeGFS共享磁碟HA叢集、藉由NetApp著作的HA元件來簡化叢集管理並提升可靠性。在NetApp上部署客製化BeeGFS架構仍是使用Ansible、在靈活的硬體範圍內維持類似應用裝置的方法。

本節概述在NetApp硬體上部署BeeGFS檔案系統、以及使用Ansible來設定BeeGFS檔案系統所需的一般步驟。如需有關 BeeGFS 檔案系統設計的最佳實務、以及最佳化範例的詳細資訊"NetApp認證的架構"、請參閱一節。

部署總覽

部署BeeGFS檔案系統通常需要執行下列步驟：

- 初始設定：
  - 安裝/纜線硬體。
  - 設定檔案和區塊節點。
  - 設定可Ansible控制節點。
- 將BeeGFS檔案系統定義為可Ansible庫存。
- 針對檔案和區塊節點執行Ansible、以部署BeeGFS。
  - （可選）設置客戶端和BeeGFS掛載。

後續章節將更詳細地說明這些步驟。

Ansible負責所有的軟體資源配置與組態工作、包括：

- 在區塊節點上建立/對應磁碟區。
- 在檔案節點上格式化/調整磁碟區。
- 在檔案節點上安裝/設定軟體。
- 建立HA叢集並設定BeeGFS資源和檔案系統服務。

需求

Ansible的BeeGFS支援已於發表 "Ansible Galaxy" 這是一套角色與模組的集合、可將BeeGFS HA叢集的端點對

端點部署與管理自動化。

BeeGFS本身的版本是根據<major> 一份《Section.Section.》<minor> 版本管理方案進行版本管理<patch> 、而該集合則負責維護<major> 每個支援的BeeGFS版本（<minor> 例如BeeGFS 7.2或BeeGFS 7.3）的角色。隨著集合更新發行、每個角色的修補程式版本將會更新、以指出該版本分支的最新可用BeeGFS版本（例如：7.2.8）。該集合的每個版本也都經過測試並支援特定的 Linux 發行版和版本，目前檔案節點使用 Red Hat，客戶端使用 Red Hat 和 Ubuntu。不支援執行其他發佈版本、不建議執行其他版本（尤其是其他主要版本）。

Ansible Control Node

此節點將包含用於管理BeeGFS的目錄和方針。它需要：

- Ansible 6.x（Ansible核心2.13）
- Python 3.6（或更新版本）
- Python（pip）套件：ipaddr和netaddr

此外、建議您從控制節點設定無密碼SSH、以連接所有BeeGFS檔案節點和用戶端。

BeeGFS檔案節點

檔案節點必須執行 Red Hat Enterprise Linux (RHEL) 9.4，並有權存取包含所需軟體包（pacemaker、corosync、fence-agents-all、resource-agents）的 HA 儲存庫。例如，可以執行下列命令在 RHEL 9 上啟用對應的儲存庫：

```
subscription-manager repo-override repo=rhel-9-for-x86_64-
highavailability-rpms --add=enabled:1
```

BeeGFS用戶端節點

BeeGFS用戶端Ansible角色可用於安裝BeeGFS用戶端套件、以及管理BeeGFS掛載。此角色已使用 RHEL 9.4 和 Ubuntu 22.04 進行測試。

如果您不使用Ansible來設定BeeGFS用戶端和BeeGFS、則可以選擇任何 "BeeGFS支援Linux發佈與核心" 可以使用。

初始設定

安裝及纜線硬體

在NetApp上安裝和連接硬體以執行BeeGFS所需的步驟。

規劃安裝

每個BeeGFS檔案系統都會包含一些使用某些區塊節點所提供的後端儲存設備執行BeeGFS服務的檔案節點。檔案節點已設定為一個或多個高可用度叢集、以提供BeeGFS服務的容錯能力。每個區塊節點都已是作用中/作用中的HA配對。每個HA叢集中支援的檔案節點數目下限為三個、每個叢集中支援的檔案節點數目上限為十個。BeeGFS檔案系統可透過部署多個獨立的HA叢集來搭配運作、以提供單一檔案系統命名空間、擴充至超過十個節點。

一般而言、每個HA叢集都會部署為一系列的「建置區塊」、其中有一些檔案節點（x86伺服器）會直接連線至某些數量的區塊節點（通常是E系列儲存系統）。此組態會建立非對稱叢集、BeeGFS服務只能在存取BeeGFS目標所用後端區塊儲存設備的特定檔案節點上執行。每個建置區塊中的檔案對區塊節點、以及直接連線所使用的儲存傳輸協定之間的平衡、取決於特定安裝的需求。

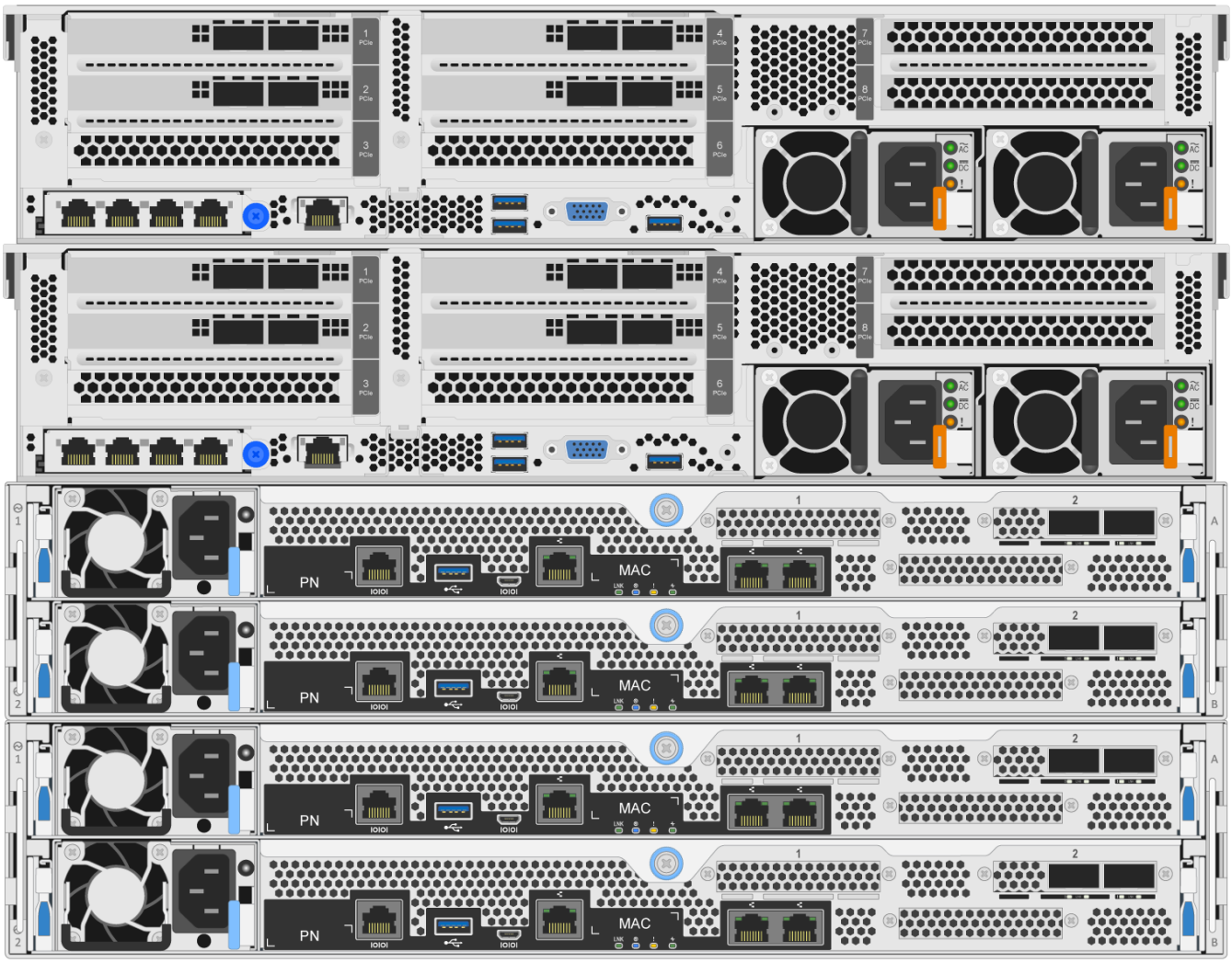
替代的HA叢集架構會在檔案和區塊節點之間使用儲存網路（也稱為儲存區域網路或SAN）來建立對稱叢集。這可讓BeeGFS服務在特定HA叢集中的任何檔案節點上執行。由於對稱叢集的成本效益通常不如額外的SAN硬體、因此本文件假設使用非對稱叢集部署為一系列一或多個建置區塊。

在繼續安裝之前、請先確認您已充分瞭解特定BeeGFS部署所需的檔案系統架構。

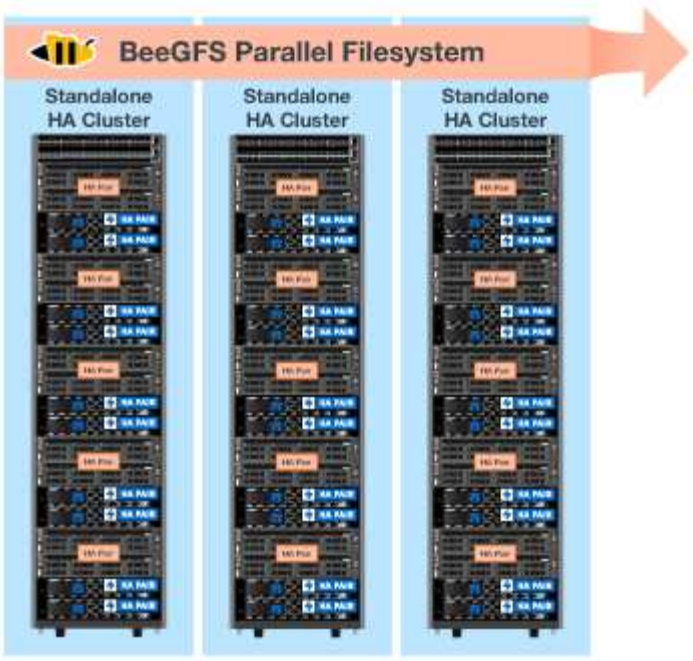
機架硬體

規劃安裝時、每個建置區塊中的所有設備都必須安裝在鄰近的機架單元中。最佳實務做法是將檔案節點直接機架在每個建置區塊的區塊節點上方。遵循檔案和模型的文件說明文件 "區塊" 將滑軌和硬體安裝到機架時所使用的節點。

單一建置區塊範例：

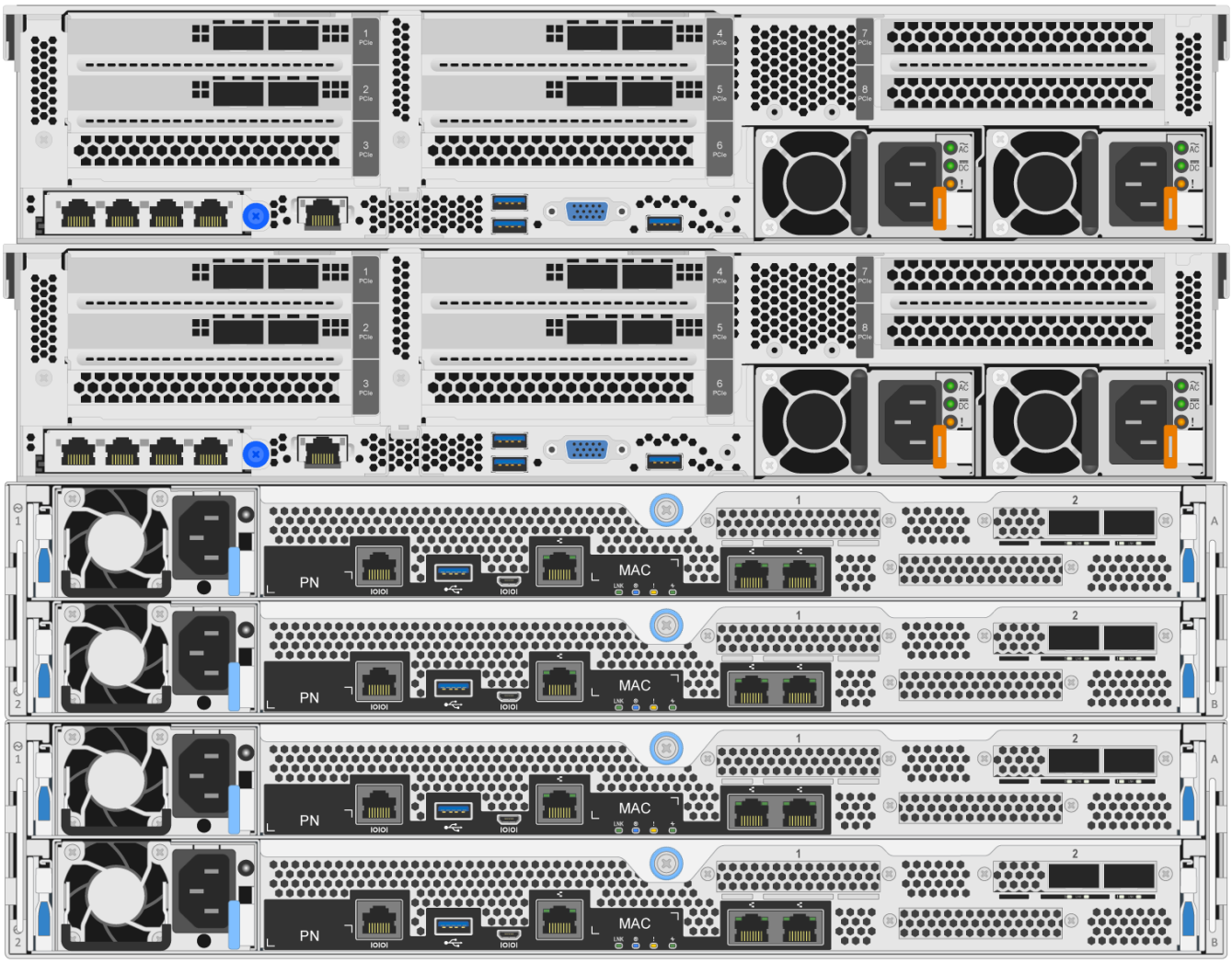


大型BeeGFS安裝範例、其中每個HA叢集有多個建置區塊、以及檔案系統中的多個HA叢集：



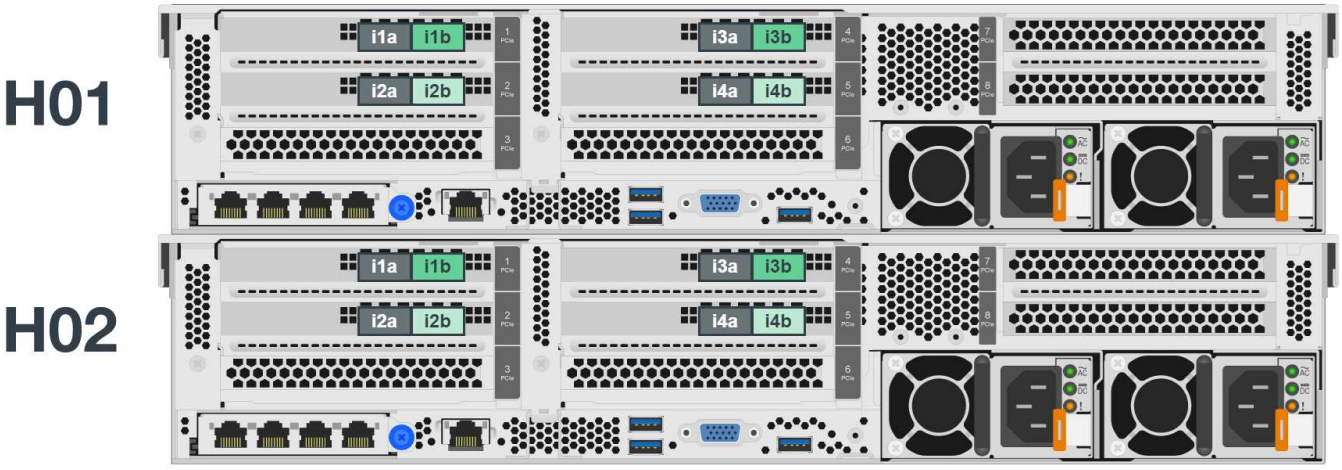
纜線檔案和區塊節點

一般而言、您會將E系列區塊節點的HIC連接埠直接連接至檔案節點的指定主機通道介面卡（適用於InfiniBand傳輸協定）或主機匯流排介面卡（適用於光纖通道和其他傳輸協定）連接埠。建立這些連線的確切方法取決於所需的檔案系統架構、以下是範例"以NetApp認證架構上的第二代BeeGFS為基礎"：



將檔案節點纜線連接至用戶端網路

每個檔案節點都會有一些InfiniBand或乙太網路連接埠、指定用於BeeGFS用戶端流量。視架構而定、每個檔案節點都會有一或多個高效能用戶端/儲存網路連線、可能會連到多個交換器以提供備援和增加頻寬。以下是使用備援網路交換器進行用戶端佈線的範例、其中以深綠色和淺綠色強調顯示的連接埠連接至不同的交換器：



連線管理網路與電源

建立頻內和頻外網路所需的任何網路連線。

連接所有電源供應器、確保每個檔案和區塊節點都能連線至多個電力分配單元、以提供備援（若有）。

設定檔案和區塊節點

在執行Ansible之前、手動設定檔案和區塊節點所需的步驟。

檔案節點

設定基礎板管理控制器（**BMC**）

底板管理控制器（BMC）有時稱為服務處理器、是內建於各種伺服器平台的頻外管理功能的一般名稱、即使作業系統未安裝或無法存取、也能提供遠端存取。廠商通常會以自己的品牌行銷這項功能。例如、在Lenovo SR665上、BMC稱為Lenovo XClarity Controller（XCC）。

請遵循伺服器廠商的文件、啟用存取此功能所需的任何授權、並確保BMC已連線至網路、並適當設定以供遠端存取。

> ⓘ 如果需要使用RedfISH的BMC型屏障、請確認已啟用RedfISH、而且可從安裝在檔案節點上的作業系統存取BMC介面。如果BMC和營運部門共用相同的實體網路介面、則網路交換器可能需要特殊組態。

調校系統設定

使用系統設定程式（BIOS / UEFI）介面、確定設定為最大化效能。確切的設定和最佳值會因使用中的伺服器機型而有所不同。提供的指南適用於"已驗證檔案節點機型"、否則請參閱伺服器廠商的文件、以及根據您的模式所提供的最佳實務做法。

安裝作業系統

根據列出的文件節點要求安裝支持的操作系統"請按這裡"。請根據您的Linux套裝作業系統、參閱下列任何其他步驟。

Red Hat

使用 Red Hat Subscription Manager 註冊並訂閱系統，以允許從官方 Red Hat 儲存庫安裝所需的軟體包，並將更新限制在支援的 Red Hat 版本上： `subscription-manager release --set=<MAJOR_VERSION>.<MINOR_VERSION>` 。有關說明，請參閱 "如何註冊及訂閱RHEL系統"和 "如何限制更新" 。

啟用包含高可用度所需套件的Red Hat儲存庫：

```
subscription-manager repo-override --repo=rhel-9-for-x86_64
-highavailability-rpms --add=enabled:1
```

設定管理網路

設定所需的任何網路介面、以允許在頻內管理作業系統。具體步驟取決於所使用的特定Linux發佈版本。

確保SSH已啟用、且所有管理介面都可從Ansible控制節點存取。

更新**HCA**和**HBA**韌體

確保所有 HBA 和 HCA 均執行中列出的支援韌體版本"NetApp 互通性對照表"、並視需要進行升級。有關 NVIDIA ConnectX 適配器的其他建議"請按這裡"，請參見。

區塊節點

請依照下列步驟執行 "使用E系列開始運作" 可在每個區塊節點控制器上設定管理連接埠、並可選擇設定每個系統的儲存陣列名稱。

除了確保所有區塊節點都可從可存取控制節點之外、沒有其他組態需要。其餘的系統組態將使用Ansible來套用/維護。

設定Ansible Control Node

設定Ansible控制節點以部署及管理檔案系統。

總覽

Ansible控制節點是用於管理叢集的實體或虛擬Linux機器。它必須符合下列要求：

- 歡迎參加 "需求"BeeGFS HA 角色、包括 Ansible 、 Python 的安裝版本、以及任何其他 Python 套件。
- 與官員會面 "Ansible控制節點需求" 包括作業系統版本。
- 可存取所有檔案和區塊節點的SSH和HTTPS。

可以找到詳細的安裝步驟"請按這裡"。

定義BeeGFS檔案系統

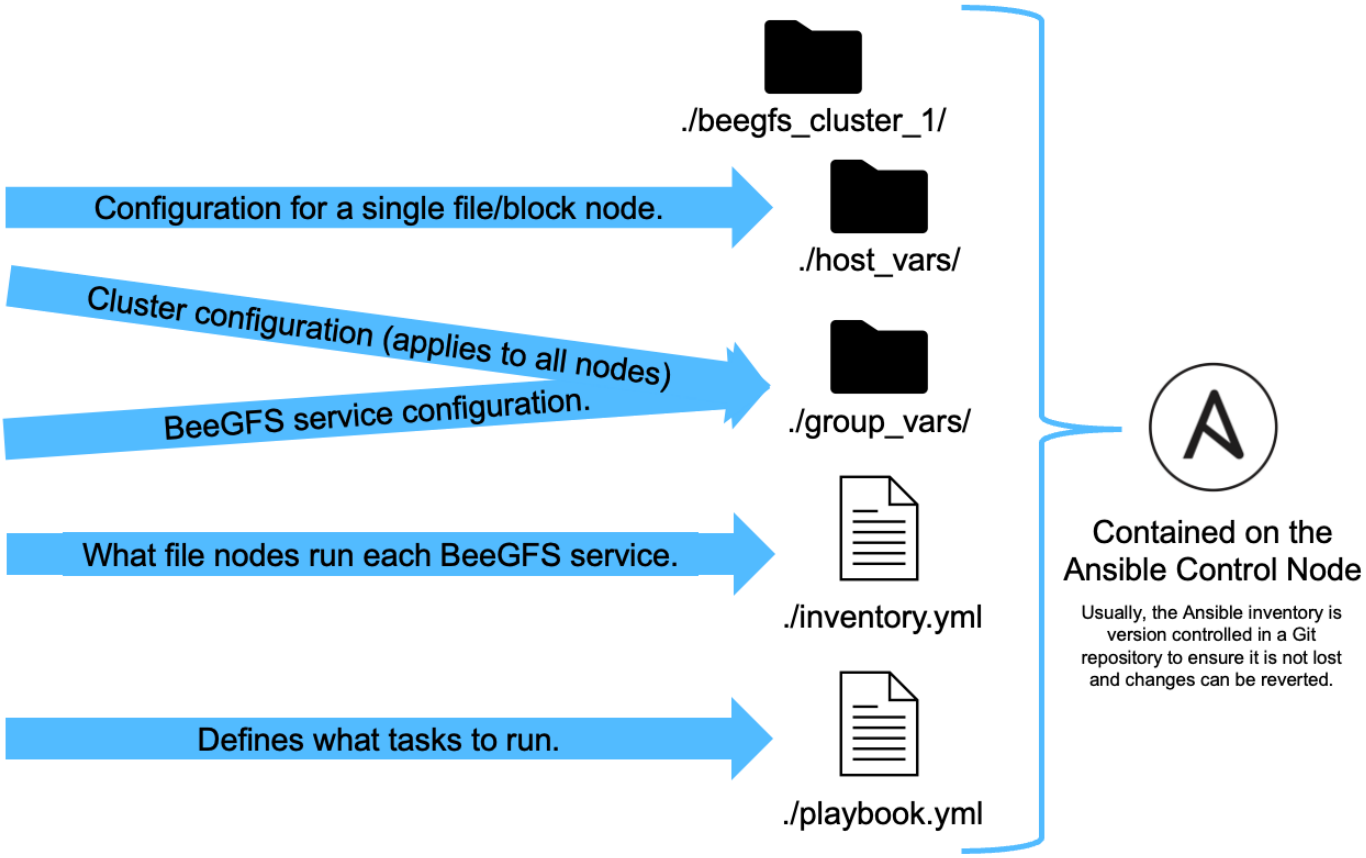
Ansible Inventory Overview

Ansible清單是一組組組態檔、可定義所需的BeeGFS HA叢集。

總覽

建議您遵循標準的可管理實務做法來組織您的 "庫存"、包括的使用 "子目錄/檔案" 而非將整個庫存儲存在單一檔案中。

單一BeeGFS HA叢集的「可安全庫存」如下所示：



由於單一BeeGFS檔案系統可橫跨多個HA叢集、因此大型安裝可能會有多個Ansible庫存。一般而言、不建議嘗試將多個HA叢集定義為單一的可Ansible庫存、以避免發生問題。

步驟

1. 在「Ansible」控制節點上、建立一個空白目錄、其中包含您要部署之BeeGFS叢集的「Ansible」清單。
   a. 如果您的檔案系統最終會包含多個HA叢集、建議您先為檔案系統建立目錄、然後為代表每個HA叢集的詳細目錄建立子目錄。例如：

```
beegfs_file_system_1/
    beegfs_cluster_1/
    beegfs_cluster_2/
    beegfs_cluster_N/
```

2. 在包含您要部署之HA叢集庫存的目錄中、建立兩個目錄 `group_vars` 和 `host_vars` 和兩個檔案 `inventory.yml` 和 `playbook.yml`。

以下各節將逐步說明這些檔案的內容定義。

規劃檔案系統

在建置Ansible庫存之前、請先規劃檔案系統部署。

總覽

在部署檔案系統之前、您應該先定義叢集中執行的所有檔案節點、區塊節點和BeeGFS服務需要哪些IP位址、連接埠和其他組態。雖然確切的組態會因叢集的架構而有所不同、但本節定義了一般適用的最佳實務做法和步驟。

步驟

1. 如果您使用IP型儲存傳輸協定（例如iSER、iSCSI、NVMe/IB或NVMe/RoCE）來將檔案節點連接至區塊節點、請填寫每個建置區塊的下列工作表。單一建置區塊中的每個直接連線都應該有唯一的子網路、而且不應與用於用戶端伺服器連線的子網路重疊。

檔案節點	IB連接埠	IP 位址	區塊節點	IB連接埠	實體IP	虛擬IP（僅適用於配備HDRIB的EF600）
<HOSTNAME>	<PORT>	<IP/SUBNET>	<HOSTNAME>	<PORT>	<IP/SUBNET>	<IP/SUBNET>

> ⓘ 如果每個建置區塊中的檔案和區塊節點是直接連線的、您通常可以針對多個建置區塊重複使用相同的IP/配置。

2. 無論您是使用InfiniBand或RDMA over Converged Ethernet（RoCE）進行儲存網路、請填寫下列工作表、以判斷將用於HA叢集服務、BeeGFS檔案服務和用戶端進行通訊的IP範圍：

目的	**InfiniBand**連接埠	**IP**位址或範圍
BeeGFS叢集IP	<INTERFACE(s)>	<RANGE>
BeeGFS管理	<INTERFACE(s)>	<IP(s)>
BeeGFS中繼資料	<INTERFACE(s)>	<RANGE>
BeeGFS儲存設備	<INTERFACE(s)>	<RANGE>
BeeGFS用戶端	<INTERFACE(s)>	<RANGE>

   a. 如果您使用單一IP子網路、則只需要一張工作表、否則請填寫第二個子網路的工作表。

3. 根據上述資訊、針對叢集中的每個建置區塊、填寫下列工作表、以定義BeeGFS將執行哪些服務。針對每項服務、指定偏好的/次要檔案節點、網路連接埠、浮動IP、NUMA區域指派（若有需要）、以及將用於其目標的區塊節點。填寫工作表時、請參閱下列準則：

   a. 也請將BeeGFS服務指定為兩者 `mgmt.yml`、 `meta_<ID>.yml`或 `storage_<ID>.yml` 其中ID代表此檔案系統中該類型所有BeeGFS服務的唯一編號。此慣例將簡化後續各節中的工作表參照、同時建立檔案以設定每項服務。

   b. BeeGFS服務的連接埠只需在特定的建置區塊中具有唯一性。請確保具有相同連接埠號碼的服務無法在同一個檔案節點上執行、以避免連接埠衝突。

   c. 必要時、服務可以使用來自多個區塊節點和（或）儲存資源池的磁碟區（並非所有磁碟區都必須由同一個控制器擁有）。多個服務也可以共用相同的區塊節點和/或儲存資源池組態（個別磁碟區將在稍後的章節中定義）。

BeeGFS服務（檔案名稱）	檔案節點	連接埠	浮動**IP**	**NUMA**區域	區塊節點	儲存資源池	擁有控制器
_setx.yml <ID> <SERVICE TYPE>	<PREFERRED FILE NODE> <SECONDARY FILE NODE(s)>	<PORT>	部分：功能<IP/SUBNET> <IP/SUBNET> <INTERFACE> <INTERFACE>	<NUMA NODE/ZONE>	<BLOCK NODE>	<STORAGE POOL/VOLUME GROUP>	<A OR B>

如需標準慣例、最佳實務做法及填寫範例工作表的詳細資訊"最佳實務做法""定義BeeGFS建置區塊"、請參閱NetApp 驗證架構上的 BeeGFS 和章節。

定義檔案和區塊節點

設定個別檔案節點

使用主機變數（host_vars）指定個別檔案節點的組態。

總覽

本節將逐步介紹填入的內容 `host_vars/<FILE_NODE_HOSTNAME>.yml` 叢集中每個檔案節點的檔案。這些檔案應僅包含特定檔案節點專屬的組態。這通常包括：

- 定義IP或Ansible主機名稱應用於連線至節點。
- 設定用於HA叢集服務（起搏器和電暈器同步）的其他介面和叢集IP、以便與其他檔案節點通訊。根據預設、這些服務使用的網路與管理介面相同、但應該有額外的介面可供備援。一般做法是在儲存網路上定義額外的IP、避免需要額外的叢集或管理網路。
  - 任何用於叢集通訊的網路效能、對檔案系統效能並不重要。使用預設叢集組態時、通常至少有 1 Gb/s 網路可為叢集作業提供足夠的效能、例如同步節點狀態和協調叢集資源狀態變更。緩慢/忙碌的網路可能會導致資源狀態變更的時間比平常長、而且在極端的情況下、如果無法在合理的時間範圍內傳送訊號、則可能會導致節點從叢集中被逐出。
- 設定介面、用於透過所需的傳輸協定連線至區塊節點（例如：iSCSI/iSER、NVMe/IB、NVMe/RoCE、FCP等）。

步驟

"規劃檔案系統"針對叢集中的每個檔案節點、參考一節中定義的 IP 定址方案會建立檔案 `host_vars/<FILE_NODE_HOSTNAME>/yml`、並填入檔案、如下所示：

1. 在頂端指定Ansible應使用的IP或主機名稱來SSH連接節點並加以管理：

```
ansible_host: "<MANAGEMENT_IP>"
```

2. 設定可用於叢集流量的其他IP：

a. 如果網路類型為 "InfiniBand（使用IPoIB）"：

```
eseries_ipoib_interfaces:
- name: <INTERFACE>  # Example: ib0 or i1b
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

b. 如果網路類型為 "融合式乙太網路上的RDMA（RoCE）"：

```
eseries_roce_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

c. 如果網路類型為 "乙太網路（僅TCP、無RDMA）"：

```
eseries_ip_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

3. 指出叢集流量應使用哪些IP、優先IP列在較高的位置：

```
beegfs_ha_cluster_node_ips:
- <MANAGEMENT_IP> # Including the management IP is typically but not
required.
- <IP_ADDRESS>    # Ex: 100.127.100.1
- <IP_ADDRESS>    # Additional IPs as needed.
```

在步驟2中設定的IPS不會作為叢集IP使用、除非包含在中 `beegfs_ha_cluster_node_ips` 清單。這可讓您使用Ansible來設定其他IP /介面、以便在需要時用於其他用途。

4. 如果檔案節點需要透過IP型傳輸協定來通訊區塊節點、則必須在適當的介面上設定IP、以及安裝/設定該傳輸協定所需的任何套件。

a. 如果使用 "iSCSI"：

```
eseries_iscsi_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
```

b. 如果使用 "商用"：

```
eseries_ib_iser_interfaces:
- name: <INTERFACE>  # Example: ib0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
  configure: true # If the file node is directly connected to the
block node set to true to setup OpenSM.
```

c. 如果使用 "NVMe / IB"：

```
eseries_nvme_ib_interfaces:
- name: <INTERFACE>  # Example: ib0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
  configure: true # If the file node is directly connected to the
block node set to true to setup OpenSM.
```

d. 如果使用 "NVMe / RoCE"：

```
eseries_nvme_roce_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
```

e. 其他通訊協定：

i. 如果使用 "NVMe / FC"，不需要設定個別介面。BeeGFS叢集部署會自動偵測傳輸協定、並視需要安裝/設定需求。如果您使用Fabric來連接檔案和區塊節點、請確保交換器已依照NetApp和交換器廠商的最佳實務做法正確分區。

ii. 使用FCP或SAS不需要安裝或設定其他軟體。如果使用FCP、請確定交換器已正確分區如下 "NetApp" 以及交換器廠商的最佳實務做法。

iii. 目前不建議使用IB SRP。視E系列區塊節點支援的項目而定、請使用NVMe/IB或iSER。

按一下 "請按這裡" 例如、代表單一檔案節點的完整庫存檔案。

進階：在乙太網路與**InfiniBand**模式之間切換**NVIDIA ConnectX VPI**介面卡

NVIDIA ConnectX-Virtual Protocol Interconnect（VPI）介面卡支援InfiniBand和乙太網路作為傳輸層。在模式之間切換不會自動協商，而且必須使用中隨附的工具進行設定 `mstconfig` `mstflint`，這是屬於的開放原始碼套件 "NVIDIA Firmare 工具（ MFT ）"。只需變更一次介面卡模式即可。這可以手動完成、也可以納入 Ansible 庫存、做為任何使用庫存區段設定的介面一部分 `eseries-`

[ib|ib_iser|ipoib|nvme_ib|nvme_roce|roce]_interfaces: 、以自動檢查 / 套用。

例如、將InfiniBand模式中的介面電流變更為乙太網路、以便用於RoCE：

1. 針對您要設定的每個介面指定 mstconfig 做為指定的對應（或字典） LINK_TYPE_P<N> 其中 <N> 由HCA的介面連接埠號碼決定。。 <N> 值可透過執行來決定 grep PCI_SLOT_NAME /sys/class/net/<INTERFACE_NAME>/device/uevent 並從PCI插槽名稱新增1至最後一個數字、然後轉換為十進位。
   a. 例如給定的 PCI_SLOT_NAME=0000:2f:00.2 （2 + 1 → HCA連接埠3）→ LINK_TYPE_P3: eth：

```
eseries_roce_interfaces:
- name: <INTERFACE>
  address: <IP/SUBNET>
  mstconfig:
    LINK_TYPE_P3: eth
```

如需其他詳細資料、請參閱 "NetApp E系列主機系列文件" 針對您使用的介面類型/傳輸協定。

設定個別區塊節點

使用主機變數（host_vars）指定個別區塊節點的組態。

總覽

本節將逐步介紹填入的內容 host_vars/<BLOCK_NODE_HOSTNAME>.yml 叢集中每個區塊節點的檔案。這些檔案應僅包含特定區塊節點專屬的組態。這通常包括：

- 系統名稱（如System Manager所示）。
- 其中一個控制器的HTTPS URL（用於使用REST API管理系統）。
- 用於連線至此區塊節點的儲存傳輸協定檔案節點。
- 設定主機介面卡（HIC）連接埠、例如IP位址（如有需要）。

步驟

"規劃檔案系統"針對叢集中的每個區塊節點、參考一節中定義的 IP 定址方案 host_vars/<BLOCK_NODE_HOSTNAME>/yml、建立檔案並填入如下內容：

1. 在頂端指定其中一個控制器的系統名稱和HTTPS URL：

```
eseries_system_name: <SYSTEM_NAME>
eseries_system_api_url:
https://<MANAGEMENT_HOSTNAME_OR_IP>:8443/devmgr/v2/
```

2. 選取 "傳輸協定" 檔案節點將用於連線至此區塊節點：
   a. 支援的傳輸協定： auto、 iscsi、 fc、 sas、 ib_srp、 ib_iser、 nvme_ib、 nvme_fc、

nvme_roce。

```
eseries_initiator_protocol: <PROTOCOL>
```

3. 視使用中的傳輸協定而定、HIC連接埠可能需要額外的組態。必要時、應定義HIC連接埠組態、使每個控制器組態的頂端項目對應於每個控制器上的實體最左側連接埠、而底部連接埠則對應最右側的連接埠。所有連接埠都需要有效的組態、即使目前未使用。

> 如果您使用的是具有EF600區塊節點的HDR（200GB）InfiniBand或200GB RoCE、請參閱以下章節。

a. 對於iSCSI：

```
eseries_controller_iscsi_port:
  controller_a:          # Ordered list of controller A channel
definition.
    - state:             # Whether the port should be enabled.
Choices: enabled, disabled
      config_method:     # Port configuration method Choices: static,
dhcp
      address:           # Port IPv4 address
      gateway:           # Port IPv4 gateway
      subnet_mask:       # Port IPv4 subnet_mask
      mtu:               # Port IPv4 mtu
    - (...)              # Additional ports as needed.
  controller_b:          # Ordered list of controller B channel
definition.
    - (...)              # Same as controller A but for controller B

# Alternatively the following common port configuration can be
defined for all ports and omitted above:
eseries_controller_iscsi_port_state: enabled         # Generally
specifies whether a controller port definition should be applied
Choices: enabled, disabled
eseries_controller_iscsi_port_config_method: dhcp    # General port
configuration method definition for both controllers. Choices:
static, dhcp
eseries_controller_iscsi_port_gateway:               # General port
IPv4 gateway for both controllers.
eseries_controller_iscsi_port_subnet_mask:           # General port
IPv4 subnet mask for both controllers.
eseries_controller_iscsi_port_mtu: 9000              # General port
maximum transfer units (MTU) for both controllers. Any value greater
than 1500 (bytes).
```

b. 對於iSER：

```
eseries_controller_ib_iser_port:
  controller_a:    # Ordered list of controller A channel address
definition.
    -              # Port IPv4 address for channel 1
    - (...)        # So on and so forth
  controller_b:    # Ordered list of controller B channel address
definition.
```

c. 適用於NVMe/IB：

```
eseries_controller_nvme_ib_port:
  controller_a:    # Ordered list of controller A channel address
definition.
    -              # Port IPv4 address for channel 1
    - (...)        # So on and so forth
  controller_b:    # Ordered list of controller B channel address
definition.
```

d. 適用於NVMe / RoCE：

```
eseries_controller_nvme_roce_port:
  controller_a:          # Ordered list of controller A channel
definition.
    - state:             # Whether the port should be enabled.
      config_method:     # Port configuration method Choices: static,
dhcp
      address:           # Port IPv4 address
      subnet_mask:       # Port IPv4 subnet_mask
      gateway:           # Port IPv4 gateway
      mtu:               # Port IPv4 mtu
      speed:             # Port IPv4 speed
  controller_b:          # Ordered list of controller B channel
definition.
    - (...)              # Same as controller A but for controller B

# Alternatively the following common port configuration can be
defined for all ports and omitted above:
eseries_controller_nvme_roce_port_state: enabled         # Generally
specifies whether a controller port definition should be applied
Choices: enabled, disabled
eseries_controller_nvme_roce_port_config_method: dhcp    # General
port configuration method definition for both controllers. Choices:
static, dhcp
eseries_controller_nvme_roce_port_gateway:               # General
port IPv4 gateway for both controllers.
eseries_controller_nvme_roce_port_subnet_mask:           # General
port IPv4 subnet mask for both controllers.
eseries_controller_nvme_roce_port_mtu: 4200              # General
port maximum transfer units (MTU). Any value greater than 1500
(bytes).
eseries_controller_nvme_roce_port_speed: auto            # General
interface speed. Value must be a supported speed or auto for
automatically negotiating the speed with the port.
```

e. FC和SAS傳輸協定不需要額外的組態。不正確建議使用SRP。

如需設定HIC連接埠和主機傳輸協定的其他選項、包括設定iSCSI CHAP的能力、請參閱 "文件" 隨附SANtricity於此系列產品。注意：部署BeeGFS時、儲存資源池、磁碟區組態及其他資源配置方面將會設定在其他位置、不應在此檔案中定義。

按一下 "請按這裡" 例如、代表單一區塊節點的完整庫存檔案。

在NetApp EF600區塊節點上使用HDR(200Gb) InfiniBand或200GB RoCE：

若要將HDR（200GB）InfiniBand搭配EF600使用、必須為每個實體連接埠設定第二個「虛擬」IP。以下是正確設定配備雙埠 InfiniBand HDR HIC 的 EF600 的範例：

```
eseries_controller_nvme_ib_port:
  controller_a:
    - 192.168.1.101    # Port 2a (virtual)
    - 192.168.2.101    # Port 2b (virtual)
    - 192.168.1.100    # Port 2a (physical)
    - 192.168.2.100    # Port 2b (physical)
  controller_b:
    - 192.168.3.101    # Port 2a (virtual)
    - 192.168.4.101    # Port 2b (virtual)
    - 192.168.3.100    # Port 2a (physical)
    - 192.168.4.100    # Port 2b (physical)
```

指定通用檔案節點組態

使用群組變數（群組_vars）指定通用檔案節點組態。

總覽

所有檔案節點的組態均應定義於 `group_vars/ha_cluster.yml`。通常包括：

- 如何連線及登入每個檔案節點的詳細資料。
- 通用網路組態。
- 是否允許自動重新開機。
- 如何設定防火牆和SELinux狀態。
- 叢集組態、包括警示和隔離。
- 效能調校：
- 通用BeeGFS服務組態。

此檔案中設定的選項也可在個別檔案節點上定義、例如使用混合式硬體模型、或是每個節點的密碼不同。個別檔案節點的組態優先於此檔案中的組態。

步驟

建立檔案 `group_vars/ha_cluster.yml` 並填入如下內容：

1. 指出Ansible Control節點應如何與遠端主機進行驗證：

```
ansible_ssh_user: root
ansible_become_password: <PASSWORD>
```

尤其是在正式作業環境中、請勿以純文字儲存密碼。請改用Ansible Vault（請參閱） "使用Ansible Vault加密內容"）或 `--ask-become-pass` 執行教戰手冊時的選項。如果是 `ansible_ssh_user` 已經是root、您可以選擇省略 `ansible_become_password`。

2. 如果您在乙太網路或InfiniBand介面上設定靜態IP（例如叢集IP）、且多個介面位於同一個IP子網路（例如、如果ib0使用192.168.1.10/24、而ib1使用192.168.1.11/24）、必須設定其他IP路由表和規則、多重主目錄支援才能正常運作。只需啟用提供的網路介面組態掛勾、如下所示：

```
eseries_ip_default_hook_templates:
  - 99-multihoming.j2
```

3. 部署叢集時、視儲存傳輸協定而定、可能需要重新啟動節點、以協助探索遠端區塊裝置（E系列磁碟區）或套用組態的其他層面。依預設、節點會在重新開機前提示、但您可以指定下列項目、讓節點自動重新啟動：

```
eseries_common_allow_host_reboot: true
```

   a. 根據預設、重新開機後、若要確保區塊裝置和其他服務已就緒、Ansible將會等到系統完成 `default.target` 在繼續部署之前就已到達。在使用NVMe / IB的某些情況下、這可能不夠長、無法初始化、探索及連線至遠端裝置。這可能會導致自動部署提早繼續、而且失敗。若要避免這種情況、使用NVMe / IB時也必須定義下列項目：

```
eseries_common_reboot_test_command: "! systemctl status
eseries_nvme_ib.service || systemctl --state=exited | grep
eseries_nvme_ib.service"
```

4. BeeGFS和HA叢集服務需要多個防火牆連接埠才能進行通訊。除非您想要手動設定防火牆（不建議）、否則請指定下列項目、以建立必要的防火牆區域並自動開啟連接埠：

```
beegfs_ha_firewall_configure: True
```

5. 目前不支援SELinux、建議將狀態設為停用、以避免衝突（尤其是使用RDMA時）。請設定下列項目、以確保SELinux已停用：

```
eseries_beegfs_ha_disable_selinux: True
eseries_selinux_state: disabled
```

6. 設定驗證、讓檔案節點能夠通訊、根據組織原則調整預設值：

```
beegfs_ha_cluster_name: hacluster                    # BeeGFS HA cluster
name.
beegfs_ha_cluster_username: hacluster                # BeeGFS HA cluster
username.
beegfs_ha_cluster_password: hapassword               # BeeGFS HA cluster
username's password.
beegfs_ha_cluster_password_sha512_salt: randomSalt # BeeGFS HA cluster
username's password salt.
```

7. 根據"規劃檔案系統"指定此檔案系統的 BeeGFS 管理 IP 一節：

```
beegfs_ha_mgmtd_floating_ip: <IP ADDRESS>
```

> 儘管看似冗餘、但當您將BeeGFS檔案系統擴充至單一HA叢集以外的位置時、「beegfs_ha_mgmtd_浮 點_ip'是很重要的。部署後續HA叢集時、不需要額外的BeeGFS管理服務、並指向第一個叢集所提供的管理服務。

8. 視需要啟用電子郵件警示：

```
beegfs_ha_enable_alerts: True
# E-mail recipient list for notifications when BeeGFS HA resources
change or fail.
beegfs_ha_alert_email_list: ["<EMAIL>"]
# This dictionary is used to configure postfix service
(/etc/postfix/main.cf) which is required to set email alerts.
beegfs_ha_alert_conf_ha_group_options:
      # This parameter specifies the local internet domain name. This is
optional when the cluster nodes have fully qualified hostnames (i.e.
host.example.com)
      mydomain: <MY_DOMAIN>
beegfs_ha_alert_verbosity: 3
#  1) high-level node activity
#  3) high-level node activity + fencing action information + resources
(filter on X-monitor)
#  5) high-level node activity + fencing action information + resources
```

9. 強烈建議啟用隔離功能、否則當主要節點故障時、服務可能無法在次要節點上啟動。

   a. 指定下列項目以全域啟用隔離：

```
beegfs_ha_cluster_crm_config_options:
  stonith-enabled: True
```

i. 附註如有需要、也可在此處指定任何支援 "叢集內容" 的項目。由於 BeeGFS HA 角色隨附許多經過測試的功能，因此通常不需要調整這些 "預設值"功能。

b. 接下來選取並設定隔離代理程式：

i. 選項1：若要使用APC電力分配單元（PDU）啟用隔離功能：

```
beegfs_ha_fencing_agents:
  fence_apc:
    - ipaddr: <PDU_IP_ADDRESS>
      login: <PDU_USERNAME>
      passwd: <PDU_PASSWORD>
      pcmk_host_map:
"<HOSTNAME>:<PDU_PORT>,<PDU_PORT>;<HOSTNAME>:<PDU_PORT>,<PDU_PORT>
"
```

ii. 選項2：若要使用Lenovo XCC（及其他BMC）提供的RedfISH API來啟用屏障：

```
redfish: &redfish
  username: <BMC_USERNAME>
  password: <BMC_PASSWORD>
  ssl_insecure: 1 # If a valid SSL certificate is not available
specify "1".

beegfs_ha_fencing_agents:
  fence_redfish:
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish
```

iii. 如需設定其他隔離代理程式的詳細資訊，請參閱 "Red Hat 文檔"。

10. BeeGFS HA角色可套用許多不同的調校參數、以協助進一步最佳化效能。其中包括最佳化核心記憶體使用率和區塊裝置I/O、以及其他參數。根據 NetApp E-Series 區塊節點的測試、角色隨附一組合理的 "預設值"、但預設不會套用這些功能、除非您指定：

```
beegfs_ha_enable_performance_tuning: True
```

a. 如有需要、也可在此處指定預設效能調校的任何變更。如需其他詳細資料、請參閱完整 "效能調校參數" 文件。

11. 為了確保BeeGFS服務所使用的浮動IP位址（有時稱為邏輯介面）可在檔案節點之間容錯移轉、所有網路介面必須一致命名。根據預設、網路介面名稱是由核心產生、因此無法保證產生一致的名稱、即使是安裝在相同PCIe插槽中的網路介面卡、也能在相同的伺服器機型上產生一致的名稱。在部署設備之前建立庫存並已知

產生介面名稱時、這也很有用。根據伺服器或的區塊圖、確保裝置名稱一致 `lshw -class network -businfo` 輸出時、請指定所需的PCIe位址對邏輯介面對應、如下所示：

a. 對於InfiniBand（IPoIB）網路介面：

```
eseries_ipoib_udev_rules:
  "<PCIe ADDRESS>": <NAME> # Ex: 0000:01:00.0: i1a
```

b. 對於乙太網路介面：

```
eseries_ip_udev_rules:
  "<PCIe ADDRESS>": <NAME> # Ex: 0000:01:00.0: e1a
```

> 為了避免在重新命名介面時發生衝突（避免重新命名）、您不應使用任何可能的預設名稱、例如eth0、ens9f0、ib0或ibs4f0。一般的命名慣例是使用「e」或「i」作為乙太網路或InfiniBand、接著是PCIe插槽編號、以及字母來表示連接埠。例如、安裝在插槽3的InfiniBand介面卡的第二個連接埠為：i3b。

> 如果您使用已驗證的檔案節點模型、請按一下 "請按這裡" PCIe位址對邏輯連接埠對應範例。

12. （可選）指定應套用至叢集中所有BeeGFS服務的組態。可以找到預設組態值 "請按這裡"、並在其他地方指定個別服務組態：

a. BeeGFS管理服務：

```
beegfs_ha_beegfs_mgmtd_conf_ha_group_options:
  <OPTION>: <VALUE>
```

b. BeeGFS中繼資料服務：

```
beegfs_ha_beegfs_meta_conf_ha_group_options:
  <OPTION>: <VALUE>
```

c. BeeGFS儲存服務：

```
beegfs_ha_beegfs_storage_conf_ha_group_options:
  <OPTION>: <VALUE>
```

13. 截至BeeGFS 7.2.7和7.3.1 "連線驗證" 必須設定或明確停用。您可以使用以Ansible為基礎的部署來設定這項功能：

a. 根據預設、部署會自動設定連線驗證、並產生 `connauthfile` 將會發佈至所有檔案節點、並搭配BeeGFS服務使用。此檔案也會放置/維護在的Ansible控制節點上

`<INVENTORY>/files/beegfs/<sysMgmtdHost>_connAuthFile` 應將其維護（安全）以供需要存取此檔案系統的用戶端重複使用。

i. 產生新的金鑰指定 `-e "beegfs_ha_conn_auth_force_new=True` 執行Ansible教戰手冊時。請注意、如果是、則會忽略此項 `beegfs_ha_conn_auth_secret` 已定義。

ii. 如需進階選項，請參閱隨附的完整預設清單 "BeeGFS HA角色"。

b. 您可以在中定義下列項目、以使用自訂密碼 `ha_cluster.yml`：

```
beegfs_ha_conn_auth_secret: <SECRET>
```

c. 連線驗證可完全停用（不建議）：

```
beegfs_ha_conn_auth_enabled: false
```

按一下 "請按這裡" 例如、代表通用檔案節點組態的完整庫存檔案。

使用具有**NetApp EF600**區塊節點的**HDR（200GB）InfiniBand**：

若要將HDR(200Gb) InfiniBand搭配EF600使用、子網路管理程式必須支援虛擬化。如果使用交換器連接檔案和區塊節點、則必須在整個Fabric的子網路管理程式上啟用此功能。

如果使用 InfiniBand 直接連接區塊和檔案節點、 `opensm`則必須在每個檔案節點上為直接連接至區塊節點的每個介面設定執行個體。這是透過指定 `configure: true`時間來完成"設定檔案節點儲存介面"的。

目前支援的 Linux 套裝作業系統隨附的收件匣版本 `opensm` 不支援虛擬化。而是必須從 NVIDIA OpenFabrics Enterprise Distribution （ OFED ）安裝和設定的版本 `opensm` 。雖然仍支援使用Ansible進行部署、但仍需執行幾個額外步驟：

1. 使用 Curl 或您想要的工具、將 NVIDIA 網站一節中所列 OpenSM 版本的套件下載 "技術需求" 到 `<INVENTORY>/packages/` 目錄中。例如：

```
curl -o packages/opensm-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
https://linux.mellanox.com/public/repo/mlnx_ofed/23.10-
3.2.2.0/rhel9.4/x86_64/opensm-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
curl -o packages/opensm-libs-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
https://linux.mellanox.com/public/repo/mlnx_ofed/23.10-
3.2.2.0/rhel9.4/x86_64/opensm-libs-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
```

2. 低於 `group_vars/ha_cluster.yml` 定義下列組態：

```
### OpenSM package and configuration information
eseries_ib_opensm_allow_upgrades: true
eseries_ib_opensm_skip_package_validation: true
eseries_ib_opensm_rhel_packages: []
eseries_ib_opensm_custom_packages:
  install:
    - files:
        add:
          "packages/opensm-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm": "/tmp/"
          "packages/opensm-libs-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm": "/tmp/"
    - packages:
        add:
          - /tmp/opensm-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
          - /tmp/opensm-libs-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
  uninstall:
    - packages:
        remove:
          - opensm
          - opensm-libs
      files:
        remove:
          - /tmp/opensm-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm
          - /tmp/opensm-libs-5.17.2.MLNX20240610.dc7c2998-
0.1.2310322.x86_64.rpm

eseries_ib_opensm_options:
  virt_enabled: "2"
```

指定通用區塊節點組態

使用群組變數（群組_vars）指定通用區塊節點組態。

總覽

所有區塊節點的組態均定義於 `group_vars/eseries_storage_systems.yml`。通常包括：

- 有關Ansible控制節點應如何連線至用作區塊節點的E系列儲存系統的詳細資料。
- 節點應該執行哪些韌體、NVSRAM/磁碟機韌體版本。
- 全域組態、包括快取設定、主機組態、以及應如何配置磁碟區的設定。

此檔案中設定的選項也可在個別區塊節點上定義、例如使用混合式硬體模型、或是每個節點的密碼不同。個別區塊節點的組態優先於此檔案中的組態。

步驟

建立檔案 `group_vars/eseries_storage_systems.yml` 並填入如下內容：

1. Ansible不會使用SSH連線至區塊節點、而是使用REST API。為了達成此目標、我們必須設定：

```
ansible_connection: local
```

2. 指定用於管理每個節點的使用者名稱和密碼。使用者名稱可以選擇性地省略（預設為admin）、否則您可以指定具有管理員權限的任何帳戶。同時指定是否應驗證或忽略SSL憑證：

```
eseries_system_username: admin
eseries_system_password: <PASSWORD>
eseries_validate_certs: false
```

不建議以純文字列出任何密碼。使用Ansible保存庫或提供 `eseries_system_password` 使用-Extra vars執行Ansible時。

3. （可選）指定應在節點上安裝哪些控制器韌體、NVSRAM/磁碟機韌體。這些項目必須下載至 `packages/` 執行Ansible之前的目錄。E系列控制器韌體和NVSRAM "請按這裡" 和磁碟機韌體 "請按這裡"：

```
eseries_firmware_firmware: "packages/<FILENAME>.dlp" # Ex.
"packages/RCB_11.80GA_6000_64cc0ee3.dlp"
eseries_firmware_nvsram: "packages/<FILENAME>.dlp" # Ex.
"packages/N6000-880834-D08.dlp"
eseries_drive_firmware_firmware_list:
  - "packages/<FILENAME>.dlp"
  # Additional firmware versions as needed.
eseries_drive_firmware_upgrade_drives_online: true # Recommended unless
BeeGFS hasn't been deployed yet, as it will disrupt host access if set
to "false".
```

如果指定此組態、Ansible將自動更新所有韌體、包括重新開機控制器（如有必要）、而不會出現其他提示。這對BeeGFS /主機I/O來說是不中斷營運的、但可能會導致效能暫時降低。

4. 調整全域系統組態預設值。此處列出的選項與值通常建議用於NetApp上的BeeGFS、但可視需要調整：

```
eseries_system_cache_block_size: 32768
eseries_system_cache_flush_threshold: 80
eseries_system_default_host_type: linux dm-mp
eseries_system_autoload_balance: disabled
eseries_system_host_connectivity_reporting: disabled
eseries_system_controller_shelf_id: 99 # Required by default.
```

5. 設定全域Volume資源配置預設值。此處列出的選項與值通常建議用於NetApp上的BeeGFS、但可視需要調整：

```
eseries_volume_size_unit: pct # Required by default. This allows volume
capacities to be specified as a percentage, simplifying putting together
the inventory.
eseries_volume_read_cache_enable: true
eseries_volume_read_ahead_enable: false
eseries_volume_write_cache_enable: true
eseries_volume_write_cache_mirror_enable: true
eseries_volume_cache_without_batteries: false
```

6. 如有需要、請依照下列最佳實務做法、調整Ansible選擇儲存資源池和磁碟區群組磁碟機的順序：
   a. 請先列出應用於管理和（或）中繼資料磁碟區的任何（可能較小的）磁碟機、然後列出儲存磁碟區。
   b. 根據磁碟櫃/磁碟機機箱機型、確保在可用磁碟機通道之間平衡磁碟機選擇順序。例如、在EF600不擴充的情況下、磁碟機0-11位於磁碟機通道1、而磁碟機12-23位於磁碟機通道。因此、平衡磁碟機選擇的策略是選擇 `disk shelf:drive` 99：0、99：23、99：1、99：22等如果有多個機箱、第一個數字代表磁碟機櫃ID。

```
# Optimal/recommended order for the EF600 (no expansion):
eseries_storage_pool_usable_drives:
"99:0,99:23,99:1,99:22,99:2,99:21,99:3,99:20,99:4,99:19,99:5,99:18,99
:6,99:17,99:7,99:16,99:8,99:15,99:9,99:14,99:10,99:13,99:11,99:12"
```

按一下 "請按這裡" 例如、代表通用區塊節點組態的完整庫存檔案。

定義BeeGFS服務

定義BeeGFS管理服務

BeeGFS服務是使用群組變數（群組_vars）進行設定。

總覽

本節將逐步說明如何定義BeeGFS管理服務。對於特定檔案系統、HA叢集中只應有一項此類型的服務。設定此服務包括定義：

- 服務類型（管理）。
- 定義任何僅應套用至此BeeGFS服務的組態。
- 設定一個或多個可連線至此服務的浮動IP（邏輯介面）。
- 指定磁碟區儲存此服務資料的位置/方式（BeeGFS管理目標）。

步驟

建立新檔案 `group_vars/mgmt.yml`、並參照"規劃檔案系統"區段、填入內容如下：

1. 指出此檔案代表BeeGFS管理服務的組態：

```
beegfs_service: management
```

2. 定義任何僅應套用至此BeeGFS服務的組態。除非您需要啟用配額、否則管理服務通常不需要此項功能、無論是否支援任何的組態參數 `beegfs-mgmtd.conf` 可隨附。請注意、下列參數會自動/在其他地方設定、不應在此處指定： `storeMgmtdDirectory`、 `connAuthFile`、 `connDisableAuthentication`、 `connInterfacesFile`和 `connNetFilterFile`。

```
beegfs_ha_beegfs_mgmtd_conf_resource_group_options:
  <beegfs-mgmt.conf:key>:<beegfs-mgmt.conf:value>
```

3. 設定其他服務和用戶端用來連線至此服務的一或多個浮動IP（這會自動設定BeeGFS） `connInterfacesFile` 選項）：

```
floating_ips:
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.
i1b:100.127.101.0/16
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```

4. 您也可以指定一或多個允許用於傳出通訊的IP子網路（這會自動設定BeeGFS） `connNetFilterFile` 選項）：

```
filter_ip_ranges:
  - <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. 指定BeeGFS管理目標、此服務將根據下列準則儲存資料：

   a. 相同的儲存資源池或磁碟區群組名稱可用於多個BeeGFS服務/目標、只要確保使用相同名稱即可 `name`、 `raid_level`、 `criteria_*`和 `common_*` 每個服務的組態（每個服務所列的磁碟區應該不同）。

   b. 磁碟區大小應指定為儲存資源池/磁碟區群組的百分比、且使用特定儲存資源池/磁碟區群組的所有服務/磁碟區的總容量不得超過100。注意使用 SSD 時、建議您在 Volume 群組中保留一些可用空間、以最大化 SSD 效能和使用壽命（按一下"請按這裡"以取得詳細資料）。

c. 按一下 "請按這裡" 以取得可用的完整組態選項清單 `eseries_storage_pool_configuration`。請注意一些選項、例如 `state`、 `host`、 `host_type`、 `workload_name`和 `workload_metadata` 而且磁碟區名稱會自動產生、不應在此處指定。

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
    eseries_storage_pool_configuration:
      - name: <NAME> # Ex: beegfs_m1_m2_m5_m6
        raid_level: <LEVEL> # One of: raid1, raid5, raid6, raidDiskPool
        criteria_drive_count: <DRIVE COUNT> # Ex. 4
        common_volume_configuration:
          segment_size_kb: <SEGMENT SIZE> # Ex. 128
        volumes:
          - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
            owning_controller: <CONTROLLER> # One of: A, B
```

按一下 "請按這裡" 例如、代表BeeGFS管理服務的完整庫存檔案。

定義**BeeGFS**中繼資料服務

BeeGFS服務是使用群組變數（群組_vars）進行設定。

總覽

本節將逐步說明如何定義BeeGFS中繼資料服務。對於特定檔案系統、HA叢集中至少應有一項此類型的服務。設定此服務包括定義：

- 服務類型（中繼資料）。
- 定義任何僅應套用至此BeeGFS服務的組態。
- 設定一個或多個可連線至此服務的浮動IP（邏輯介面）。
- 指定磁碟區儲存此服務資料的位置/方式（BeeGFS中繼資料目標）。

步驟

參照"規劃檔案系統"區段、 `group_vars/meta_<ID>.yml`為叢集中的每個中繼資料服務建立位於的檔案、並填入這些檔案、如下所示：

1. 指出此檔案代表BeeGFS中繼資料服務的組態：

```
beegfs_service: metadata
```

2. 定義任何僅應套用至此BeeGFS服務的組態。您至少必須指定所需的TCP和udp連接埠、無論是否支援任何的組態參數 `beegfs-meta.conf` 也可隨附。請注意、下列參數會自動/在其他地方設定、不應在此處指定：`sysMgmtdHost`、 `storeMetaDirectory`、 `connAuthFile`、 `connDisableAuthentication`、

connInterfacesFile`和 `connNetFilterFile。

```
beegfs_ha_beegfs_meta_conf_resource_group_options:
  connMetaPortTCP: <TCP PORT>
  connMetaPortUDP: <UDP PORT>
  tuneBindToNumaZone: <NUMA ZONE> # Recommended if using file nodes with
multiple CPU sockets.
```

3. 設定其他服務和用戶端用來連線至此服務的一或多個浮動IP（這會自動設定BeeGFS）connInterfacesFile 選項）：

```
floating_ips:
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.
i1b:100.127.101.1/16
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```

4. 您也可以指定一或多個允許用於傳出通訊的IP子網路（這會自動設定BeeGFS） connNetFilterFile 選項）：

```
filter_ip_ranges:
  - <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. 指定BeeGFS中繼資料目標、此服務會根據下列準則來儲存資料（這也會自動設定 storeMetaDirectory 選項）：

   a. 相同的儲存資源池或磁碟區群組名稱可用於多個BeeGFS服務/目標、只要確保使用相同名稱即可 name、 raid_level、 criteria_*`和 `common_* 每個服務的組態（每個服務所列的磁碟區應該不同）。

   b. 磁碟區大小應指定為儲存資源池/磁碟區群組的百分比、且使用特定儲存資源池/磁碟區群組的所有服務/磁碟區的總容量不得超過100。注意使用 SSD 時、建議您在 Volume 群組中保留一些可用空間、以最大化 SSD 效能和使用壽命（按一下"請按這裡"以取得詳細資料）。

   c. 按一下 "請按這裡" 以取得可用的完整組態選項清單 eseries_storage_pool_configuration。請注意一些選項、例如 state、 host、 host_type、 workload_name`和 `workload_metadata 而且磁碟區名稱會自動產生、不應在此處指定。

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
    eseries_storage_pool_configuration:
      - name: <NAME> # Ex: beegfs_m1_m2_m5_m6
        raid_level: <LEVEL> # One of: raid1, raid5, raid6, raidDiskPool
        criteria_drive_count: <DRIVE COUNT> # Ex. 4
        common_volume_configuration:
          segment_size_kb: <SEGMENT SIZE> # Ex. 128
        volumes:
          - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
            owning_controller: <CONTROLLER> # One of: A, B
```

按一下 "請按這裡" 例如、代表BeeGFS中繼資料服務的完整庫存檔案。

定義**BeeGFS**儲存服務

BeeGFS服務是使用群組變數（群組_vars）進行設定。

總覽

本節將逐步說明如何定義BeeGFS儲存服務。對於特定檔案系統、HA叢集中至少應有一項此類型的服務。設定此服務包括定義：

- 服務類型（儲存設備）。
- 定義任何僅應套用至此BeeGFS服務的組態。
- 設定一個或多個可連線至此服務的浮動IP（邏輯介面）。
- 指定磁碟區應儲存此服務資料的位置/方式（BeeGFS儲存目標）。

步驟

請參考本"規劃檔案系統"節、 `group_vars/stor_<ID>.yml`為叢集中的每個儲存服務建立位於的檔案、並依照下列步驟填入：

1. 指出此檔案代表BeeGFS儲存服務的組態：

```
beegfs_service: storage
```

2. 定義任何僅應套用至此BeeGFS服務的組態。您至少必須指定所需的TCP和udp連接埠、無論是否支援任何的組態參數 `beegfs-storage.conf` 也可隨附。請注意、下列參數會自動/在其他地方設定、不應在此處指定： `sysMgmtdHost`、 `storeStorageDirectory`、 `connAuthFile`、 `connDisableAuthentication`、 `connInterfacesFile`和 `connNetFilterFile`。

```
beegfs_ha_beegfs_storage_conf_resource_group_options:
  connStoragePortTCP: <TCP PORT>
  connStoragePortUDP: <UDP PORT>
  tuneBindToNumaZone: <NUMA ZONE> # Recommended if using file nodes with
multiple CPU sockets.
```

3. 設定其他服務和用戶端用來連線至此服務的一或多個浮動IP（這會自動設定BeeGFS） `connInterfacesFile` 選項）：

```
floating_ips:
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.
i1b:100.127.101.1/16
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```

4. 您也可以指定一或多個允許用於傳出通訊的IP子網路（這會自動設定BeeGFS） `connNetFilterFile` 選項）：

```
filter_ip_ranges:
  - <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. 指定此服務將根據下列準則儲存資料的BeeGFS儲存目標（這也會自動設定） `storeStorageDirectory` 選項）：

   a. 相同的儲存資源池或磁碟區群組名稱可用於多個BeeGFS服務/目標、只要確保使用相同名稱即可 `name`、 `raid_level`、 `criteria_*`和 `common_*` 每個服務的組態（每個服務所列的磁碟區應該不同）。

   b. 磁碟區大小應指定為儲存資源池/磁碟區群組的百分比、且使用特定儲存資源池/磁碟區群組的所有服務/磁碟區的總容量不得超過100。注意使用 SSD 時、建議您在 Volume 群組中保留一些可用空間、以最大化 SSD 效能和使用壽命（按一下"請按這裡"以取得詳細資料）。

   c. 按一下 "請按這裡" 以取得可用的完整組態選項清單 `eseries_storage_pool_configuration`。請注意一些選項、例如 `state`、 `host`、 `host_type`、 `workload_name`和 `workload_metadata` 而且磁碟區名稱會自動產生、不應在此處指定。

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
    eseries_storage_pool_configuration:
      - name: <NAME> # Ex: beegfs_s1_s2
        raid_level: <LEVEL> # One of: raid1, raid5, raid6,
raidDiskPool
        criteria_drive_count: <DRIVE COUNT> # Ex. 4
        common_volume_configuration:
          segment_size_kb: <SEGMENT SIZE> # Ex. 128
        volumes:
          - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
            owning_controller: <CONTROLLER> # One of: A, B
        # Multiple storage targets are supported / typical:
          - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
            owning_controller: <CONTROLLER> # One of: A, B
```

按一下 "請按這裡" 例如、代表BeeGFS儲存服務的完整庫存檔案。

將BeeGFS服務對應至檔案節點

使用指定可以執行每個BeeGFS服務的檔案節點 `inventory.yml` 檔案：

總覽

本節將逐步說明如何建立 `inventory.yml` 檔案：這包括列出所有區塊節點、並指定可執行每個BeeGFS服務的檔案節點。

步驟

建立檔案 `inventory.yml` 並填入如下內容：

1. 從檔案頂端建立標準的可Ansible庫存結構：

```
# BeeGFS HA (High Availability) cluster inventory.
all:
  children:
```

2. 建立一個群組、其中包含參與此HA叢集的所有區塊節點：

```
# Ansible group representing all block nodes:
eseries_storage_systems:
  hosts:
    <BLOCK NODE HOSTNAME>:
    <BLOCK NODE HOSTNAME>:
    # Additional block nodes as needed.
```

3. 建立一個群組、其中包含叢集中的所有BeeGFS服務、以及執行這些服務的檔案節點：

```
# Ansible group representing all file nodes:
ha_cluster:
  children:
```

4. 針對叢集中的每個BeeGFS服務、定義應該執行該服務的慣用和任何次要檔案節點：

```
    <SERVICE>: # Ex. "mgmt", "meta_01", or "stor_01".
      hosts:
        <FILE NODE HOSTNAME>:
        <FILE NODE HOSTNAME>:
        # Additional file nodes as needed.
```

按一下 "請按這裡" 以取得完整庫存檔案的範例。

部署BeeGFS檔案系統

Ansible教戰手冊總覽

使用Ansible部署及管理BeeGFS HA叢集。

總覽

前幾節將逐步說明如何建立代表BeeGFS HA叢集的Ansible庫存。本節將介紹NetApp開發的Ansible自動化功能、以部署及管理叢集。

Ansible：重要概念

在繼續之前、熟悉幾個重要的可執行概念是很有幫助的：

- 根據可執行的庫存執行的工作是在稱為*教戰手冊*的內容中定義。
  - Ansible中的大多數工作都是*冪等*的、這表示可以執行多次、以驗證所需的組態/狀態是否仍在套用、而不會造成任何中斷或進行不必要的更新。
- Ansible中最小的執行單位是*模組*。

    - 典型的教戰手冊使用多個模組。
        - 範例：下載套件、更新組態檔、啟動/啟用服務。
    - NetApp發佈模組以自動化NetApp E系列系統。
- 複雜的自動化功能更適合當作角色來進行套裝。
    - 基本上是一種標準格式、可用來發佈可重複使用的教戰手冊。
    - NetApp負責分配Linux主機和BeeGFS檔案系統的角色。

BeeGFS HA Ansible角色：重要概念

在NetApp上部署及管理每個版本BeeGFS所需的所有自動化作業、均以Ansible角色進行封裝、並隨附於一起散佈 "NetApp E系列BeeGFS適用的Ansible收藏"：

- 此角色可視為BeeGFS *安裝程式*與現代*部署/管理*引擎之間的某個位置。
    - 將現代化的基礎架構套用為程式碼實務做法和理念、以簡化任何規模的儲存基礎架構管理。
    - 類似"Kubespray"專案如何讓使用者部署 / 維護整個 Kubernetes 發佈、以進行橫向擴充運算基礎架構。
- 此角色是*軟體定義*格式的NetApp用於在NetApp解決方案上封裝、發佈及維護BeeGFS。
    - 努力創造「類似應用裝置」的體驗、而不需要散佈整個Linux發行版本或大型映像。
    - 包括NetApp著作的開放式叢集架構（OCF）相容叢集資源代理程式、可用於自訂BeeGFS目標、IP位址及監控、以提供智慧型起搏器/ BeeGFS整合功能。
- 此角色不只是部署「自動化」、旨在管理整個檔案系統生命週期、包括：
    - 套用個別服務或整個叢集的組態變更與更新。
    - 在硬體問題解決後、將叢集修復與還原自動化。
    - 利用BeeGFS與NetApp磁碟區的廣泛測試、設定預設值、簡化效能調校。
    - 驗證及修正組態飄移。

NetApp也為提供Ansible角色 "BeeGFS用戶端"（可選）用於安裝BeeGFS並將文件系統掛載到compute（計算）/GPU/Login（計算/ GPU /登錄）節點。

部署BeeGFS HA叢集

使用方針來指定部署BeeGFS HA叢集時應執行哪些工作。

總覽

本節說明如何組裝標準方針、以在NetApp上部署/管理BeeGFS。

步驟

建立可執行的教戰手冊

建立檔案 `playbook.yml` 並填入如下內容：

1. 首先定義一組工作（通常稱為 "玩遊戲"）、只能在NetApp E系列區塊節點上執行。我們會使用暫停工作在執

行安裝之前先行提示（以避免意外執行教戰手冊）、然後匯入 `nar_santricity_management` 角色：此角色負責套用中定義的任何一般系統組態 `group_vars/eseries_storage_systems.yml` 或個人 `host_vars/<BLOCK NODE>.yml` 檔案：

```
- hosts: eseries_storage_systems
  gather_facts: false
  collections:
    - netapp_eseries.santricity
  tasks:
    - name: Verify before proceeding.
      pause:
        prompt: "Are you ready to proceed with running the BeeGFS HA
role? Depending on the size of the deployment and network performance
between the Ansible control node and BeeGFS file and block nodes this
can take awhile (10+ minutes) to complete."
    - name: Configure NetApp E-Series block nodes.
      import_role:
        name: nar_santricity_management
```

2. 定義要在所有檔案和區塊節點上執行的播放：

```
- hosts: all
  any_errors_fatal: true
  gather_facts: false
  collections:
    - netapp_eseries.beegfs
```

3. 在這場活動中、我們可以選擇定義一組「預先工作」、在部署HA叢集之前應先執行。這對於驗證/安裝任何先決條件（如Python）來說都很有用。我們也可以進行任何飛行前檢查、例如驗證是否支援提供的Ansible標記：

```
  pre_tasks:
    - name: Ensure a supported version of Python is available on all
file nodes.
      block:
        - name: Check if python is installed.
          failed_when: false
          changed_when: false
          raw: python --version
          register: python_version

        - name: Check if python3 is installed.
          raw: python3 --version
          failed_when: false
```

```
          changed_when: false
          register: python3_version
          when: 'python_version["rc"] != 0 or (python_version["stdout"]
| regex_replace("Python ", "")) is not version("3.0", ">=")'

        - name: Install python3 if needed.
          raw: |
            id=$(grep "^ID=" /etc/*release* | cut -d= -f 2 | tr -d '"')
            case $id in
              ubuntu) sudo apt install python3 ;;
              rhel|centos) sudo yum -y install python3 ;;
              sles) sudo zypper install python3 ;;
            esac
          args:
            executable: /bin/bash
          register: python3_install
          when: python_version['rc'] != 0 and python3_version['rc'] != 0
          become: true

        - name: Create a symbolic link to python from python3.
          raw: ln -s /usr/bin/python3 /usr/bin/python
          become: true
          when: python_version['rc'] != 0
      when: inventory_hostname not in
groups[beegfs_ha_ansible_storage_group]

    - name: Verify any provided tags are supported.
      fail:
        msg: "{{ item }} tag is not a supported BeeGFS HA tag. Rerun
your playbook command with --list-tags to see all valid playbook tags."
      when: 'item not in ["all", "storage", "beegfs_ha",
"beegfs_ha_package", "beegfs_ha_configure",
"beegfs_ha_configure_resource", "beegfs_ha_performance_tuning",
"beegfs_ha_backup", "beegfs_ha_client"]'
      loop: "{{ ansible_run_tags }}"
```

4. 最後、這場活動會將BeeGFS HA角色匯入您要部署的BeeGFS版本：

```
  tasks:
    - name: Verify the BeeGFS HA cluster is properly deployed.
      import_role:
        name: beegfs_ha_7_4 # Alternatively specify: beegfs_ha_7_3.
```

每個支援的主要.次要 版本BeeGFS均維持BeeGFS HA角色。這可讓使用者選擇何時升級主要/次要版本。目前(`beegfs_7_3`(`beegfs_7_2`支援 BeeGFS 7.3.x 或 BeeGFS 7.2.x ）。根據預設、兩個角色都會在發行時部署最新的BeeGFS修補程式版本、不過使用者可以選擇覆寫此版本、並視需要部署最新的修補程式。如"升級指南"需詳細資訊、請參閱最新資訊。

5. 選用：如果您想要定義其他工作、請記住是否應將工作導向 `all` 主機（包括E系列儲存系統）或僅檔案節點。如有需要、請使用定義專為檔案節點而設計的新遊戲 `- hosts: ha_cluster`。

按一下 "請按這裡" 例如完整的教戰手冊檔案。

安裝NetApp Ansible Collection

BeeGFS的Ansible及所有相依項目集合均保留在上 "Ansible Galaxy"。在Ansible控制節點上執行下列命令、以安裝最新版本：

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

雖然通常不建議使用、但也可以安裝特定版本的集合：

```
ansible-galaxy collection install netapp_eseries.beegfs:
==<MAJOR>.<MINOR>.<PATCH>
```

執行教戰手冊

從包含的Ansible控制節點上的目錄 `inventory.yml` 和 `playbook.yml` 檔案、請依照下列步驟執行方針：

```
ansible-playbook -i inventory.yml playbook.yml
```

根據叢集的大小、初始部署可能需要20分鐘以上的時間。如果部署因任何原因而失敗、只要修正任何問題（例如：錯誤佈線、節點未啟動等）、然後重新啟動可執行的方針即可。

指定"通用檔案節點組態"時，如果您選擇預設選項讓 Ansible 自動管理連線型驗證，則 `connAuthFile`可在 `<playbook_dir>/files/beegfs/<sysMgmtdHost>_connAuthFile`（預設）找到用作共用密碼的。任何需要存取檔案系統的用戶端都必須使用此共用密碼。如果使用設定用戶端，則會自動處理此"BeeGFS用戶端角色"問題。

部署BeeGFS用戶端

（可選）Ansible可用於配置BeeGFS客戶端並掛載文件系統。

總覽

存取BeeGFS檔案系統時、必須在需要掛載檔案系統的每個節點上安裝及設定BeeGFS用戶端。本節說明如何使用可用的執行這些工作 "Ansible角色"。

步驟

建立用戶端庫存檔案

1. 如有需要、請從Ansible控制節點設定無密碼SSH、並將其設定為BeeGFS用戶端的每個主機：

```
ssh-copy-id <user>@<HOSTNAME_OR_IP>
```

2. 低於 `host_vars/`、為每個BeeGFS用戶端建立一個名為的檔案 `<HOSTNAME>.yml` 在下列內容中、以適合您環境的正確資訊填寫預留位置文字：

```
# BeeGFS Client
ansible_host: <MANAGEMENT_IP>
```

3. 如果您想要使用NetApp E系列主機集合的角色來設定InfiniBand或乙太網路介面、以便用戶端連線至BeeGFS檔案節點、也可以選擇加入下列其中一項：

   a. 如果網路類型為 "InfiniBand（使用IPoIB）"：

```
eseries_ipoib_interfaces:
- name: <INTERFACE>  # Example: ib0 or i1b
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

   b. 如果網路類型為 "融合式乙太網路上的RDMA（RoCE）"：

```
eseries_roce_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

   c. 如果網路類型為 "乙太網路（僅TCP、無RDMA）"：

```
eseries_ip_interfaces:
- name: <INTERFACE>  # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

4. 建立新檔案 `client_inventory.yml` 並指定Ansible使用者應使用連線至每個用戶端、而Ansible密碼應用於權限提升（這需要 `ansible_ssh_user` 為root、或具有Sudo權限）：

```
# BeeGFS client inventory.
all:
  vars:
    ansible_ssh_user: <USER>
    ansible_become_password: <PASSWORD>
```

請勿以純文字儲存密碼。請改用Ansible Vault（請參閱 "Ansible文件" 使用Ansible Vault加密內容）或使用 `--ask-become-pass` 執行教戰手冊時的選項。

5. 在中 `client_inventory.yml` 檔案中、列出應在中設定為BeeGFS用戶端的所有主機 `beegfs_clients` 然後參閱內嵌註解、取消註釋在系統上建置BeeGFS用戶端核心模組所需的任何其他組態：

```
children:
    # Ansible group representing all BeeGFS clients:
    beegfs_clients:
      hosts:
        <CLIENT HOSTNAME>:
        # Additional clients as needed.

      vars:
        # OPTION 1: If you're using the NVIDIA OFED drivers and they are
already installed:
        #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.
        #beegfs_client_ofed_enable: True
        #beegfs_client_ofed_include_path:
"/usr/src/ofa_kernel/default/include"

        # OPTION 2: If you're using inbox IB/RDMA drivers and they are
already installed:
        #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.

        # OPTION 3: If you want to use inbox IB/RDMA drivers and need
them installed/configured.
        #eseries_ib_skip: False # Default value.
        #beegfs_client_ofed_enable: False # Default value.
```

使用 NVIDIA OFED 驅動程式時、請確定 beegfs_client_of_ofed_include_path 指向 Linux 安裝的正確「 header include path 」。如需詳細資訊，請參閱的 BeeGFS 文件 "RDMA支援"。

6. 在中 `client_inventory.yml` 檔案中、列出您要掛載於任何先前定義下的BeeGFS檔案系統 `vars`：

```
        beegfs_client_mounts:
          - sysMgmtdHost: <IP ADDRESS>  # Primary IP of the BeeGFS
management service.
            mount_point: /mnt/beegfs    # Path to mount BeeGFS on the
client.
            connInterfaces:
              - <INTERFACE> # Example: ibs4f1
              - <INTERFACE>
            beegfs_client_config:
              # Maximum number of simultaneous connections to the same
node.
              connMaxInternodeNum: 128 # BeeGFS Client Default: 12
              # Allocates the number of buffers for transferring IO.
              connRDMABufNum: 36 # BeeGFS Client Default: 70
              # Size of each allocated RDMA buffer
              connRDMABufSize: 65536 # BeeGFS Client Default: 8192
              # Required when using the BeeGFS client with the shared-
disk HA solution.
              # This does require BeeGFS targets be mounted in the
default "sync" mode.
              # See the documentation included with the BeeGFS client
role for full details.
              sysSessionChecksEnabled: false
        # Specify additional file system mounts for this or other file
systems.
```

7. "連線驗證"必須設定或明確停用 BeeGFS 7.2.7 和 7.3.1 。根據您在指定時選擇如何設定連線型驗證"通用檔案節點組態"、您可能需要調整用戶端組態：

   a. 依預設、HA叢集部署會自動設定連線驗證、並產生 `connauthfile` 將放置/維護在的Ansible控制節點上 `<INVENTORY>/files/beegfs/<sysMgmtdHost>_connAuthFile`。根據預設、BeeGFS用戶端角色會設定為將此檔案讀取/散佈到中定義的用戶端 `client_inventory.yml`、而且不需要採取其他行動。

      i. 如需進階選項、請參閱隨附的完整預設清單 "BeeGFS用戶端角色"。

   b. 如果您選擇使用來指定自訂密碼 `beegfs_ha_conn_auth_secret` 在中指定 `client_inventory.yml` 檔案也包括：

```
beegfs_ha_conn_auth_secret: <SECRET>
```

   c. 如果您選擇完全停用以連線為基礎的驗證 `beegfs_ha_conn_auth_enabled`、在中指定 `client_inventory.yml` 檔案也包括：

```
beegfs_ha_conn_auth_enabled: false
```

如需支援參數的完整清單及其他詳細資料、請參閱 "完整的BeeGFS用戶端文件"。如需用戶端庫存的完整範例、請按一下 "請按這裡"。

建立BeeGFS用戶端教戰手冊檔案

1. 建立新檔案 `client_playbook.yml`

```
# BeeGFS client playbook.
- hosts: beegfs_clients
  any_errors_fatal: true
  gather_facts: true
  collections:
    - netapp_eseries.beegfs
    - netapp_eseries.host
  tasks:
```

2. 選用：如果您想要使用NetApp E系列主機集合的角色來設定介面、讓用戶端連線至BeeGFS檔案系統、請匯入與您所設定介面類型對應的角色：

   a. 如果您使用的是InfiniBand（IPoIB）：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: ipoib
```

   b. 如果您使用的是透過整合式乙太網路（RoCE）的RDMA：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: roce
```

   c. 如果您使用的是乙太網路（僅TCP、無RDMA）：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: ip
```

3. 最後匯入BeeGFS用戶端角色、以安裝用戶端軟體並設定檔案系統掛載：

```
    # REQUIRED: Install the BeeGFS client and mount the BeeGFS file
system.
    - name: Verify the BeeGFS clients are configured.
      import_role:
        name: beegfs_client
```

如需用戶端方針的完整範例、請按一下 "請按這裡"。

執行BeeGFS用戶端教戰手冊

若要安裝/建置用戶端及Mount BeeGFS、請執行下列命令：

```
ansible-playbook -i client_inventory.yml client_playbook.yml
```

驗證BeeGFS部署

將系統投入正式作業之前、請先確認檔案系統部署。

總覽

將BeeGFS檔案系統置於正式作業環境之前、請先執行幾項驗證檢查。

步驟

1. 登入任何用戶端並執行下列作業、以確保所有預期節點都存在/可連線、而且不會報告不一致或其他問題：

```
beegfs-fsck --checkfs
```

2. 關閉整個叢集、然後重新啟動。從任何檔案節點執行下列作業：

```
pcs cluster stop --all # Stop the cluster on all file nodes.
pcs cluster start --all  # Start the cluster on all file nodes.
pcs status # Verify all nodes and services are started and no failures
are reported (the command may need to be reran a few times to allow time
for all services to start).
```

3. 將每個節點置於待命狀態、並確認BeeGFS服務能夠容錯移轉至次要節點。若要登入任何檔案節點、並執行下列步驟：

```
pcs status # Verify the cluster is healthy at the start.
pcs node standby <FILE NODE HOSTNAME> # Place the node under test in
standby.
pcs status # Verify services are started on a secondary node and no
failures are reported.
pcs node unstandby <FILE NODE HOSTNAME> # Take the node under test out
of standby.
pcs status # Verify the file node is back online and no failures are
reported.
pcs resource relocate run # Move all services back to their preferred
nodes.
pcs status # Verify services have moved back to the preferred node.
```

4. 使用IOR和MDTest等效能基準測試工具來驗證檔案系統效能是否符合預期。在 "設計驗證"NetApp 驗證架構的 BeeGFS 一節中、可以找到 BeeGFS 使用的一般測試和參數範例。

應根據針對特定站台/安裝所定義的驗收條件來執行其他測試。

部署功能和集成

BeeGFS CSI 驅動程式

為 BeeGFS v8 配置 TLS 加密

配置 TLS 加密以保護 BeeGFS v8 管理服務和用戶端之間的通訊。

總覽

BeeGFS v8 引入了 TLS 支援，用於加密管理工具（例如 `beegfs` 命令列實用程式）與 BeeGFS 伺服器服務（例如 Management 或 Remote）之間的網路通訊。本指南介紹如何使用三種 TLS 設定方法在 BeeGFS 叢集中設定 TLS 加密：

- 使用受信任的憑證授權單位：在您的 BeeGFS 叢集上使用現有的 CA 簽署憑證。
- 建立本地憑證授權單位： 建立本地憑證授權單位並使用它來簽署 BeeGFS 服務的憑證。這種方法適用於您希望管理自己的信任鏈而不依賴外部 CA 的環境。
- **TLS** 已停用： 在不需要加密的環境或進行故障排除時，可以完全停用 TLS。不建議這樣做，因為它會將內部檔案系統結構和配置等潛在敏感資訊以明文形式暴露出來。

選擇最適合您環境和組織政策的方法。請參閱 "BeeGFS TLS" 文件以獲取更多詳細資訊。

> 運行 `beegfs-client`服務的機器無需 TLS 即可掛載 BeeGFS 檔案系統。必須設定 TLS 才能使用 BeeGFS CLI 和其他 beegfs 服務，例如 remote 和 sync。

使用受信任的憑證授權單位

如果您可以存取受信任的憑證授權單位 (CA) 所頒發的憑證（無論是來自企業內部 CA 還是第三方提供者），您可以設定 BeeGFS v8 使用這些 CA 簽署的憑證，而不是產生自簽名憑證。

部署新的 **BeeGFS v8** 叢集

對於新的 BeeGFS v8 叢集部署，請設定 Ansible 清單的 `user_defined_params.yml` 檔案以引用您的 CA 簽署憑證：

```
beegfs_ha_tls_enabled: true

beegfs_ha_ca_cert_src_path: files/beegfs/cert/ca_cert.pem

beegfs_ha_tls_cert_src_path: files/beegfs/cert/mgmtd_tls_cert.pem

beegfs_ha_tls_key_src_path: files/beegfs/cert/mgmtd_tls_key.pem
```

如果 `beegfs_ha_tls_config_options.alt_names`不為空，Ansible 會自動產生自簽名 TLS 憑證和金鑰，並使用提供的 alt_names 作為憑證中的主題備用名稱（SAN）。若要使用您自己的自訂 TLS 憑證和金鑰（如 `beegfs_ha_tls_cert_src_path`和 `beegfs_ha_tls_key_src_path`所指定），您必須註解掉或刪除整個 `beegfs_ha_tls_config_options`部分。否則，自簽名憑證的產生將優先，您的自訂憑證和金鑰將不會被使用。

配置現有的 **BeeGFS v8** 叢集

對於現有的 BeeGFS v8 集群，請將 BeeGFS 管理服務的設定檔中的路徑設定為檔案節點的 CA 簽章憑證：

```
tls-cert-file = /path/to/cert.pem
tls-key-file = /path/to/key.pem
```

使用 **CA** 簽署憑證設定 **BeeGFS v8** 用戶端

若要設定 BeeGFS v8 用戶端以信任使用系統憑證池的 CA 簽章證書，請在每個用戶端的設定中設定 tls-cert-file = ""。如果未使用系統憑證池，請透過設定 tls-cert-file = <local cert> 來提供本機憑證的路徑。此設定允許用戶端驗證 BeeGFS 管理服務提供的憑證。

建立本地憑證授權單位

如果您的組織希望為 BeeGFS 叢集建立自己的憑證基礎架構，您可以建立一個本機憑證授權單位（CA）來頒發和簽署 BeeGFS 叢集的憑證。此方法涉及建立一個 CA，該 CA 為 BeeGFS 管理服務簽署憑證，然後將這些憑證分發給客戶端以建立信任鏈。請依照下列說明設定本機 CA 並在現有或新的 BeeGFS v8 叢集上部署憑證。

部署新的 **BeeGFS v8** 叢集

對於新的 BeeGFS v8 部署， `beegfs_8` Ansible 角色將負責在控制節點上建立本機 CA，並為管理服務產生必要的憑證。可以透過在 Ansible 清單的`user_defined_params.yml`檔案中設定以下參數來啟用此功能：

```
beegfs_ha_tls_enabled: true

beegfs_ha_ca_cert_src_path: files/beegfs/cert/local_ca_cert.pem

beegfs_ha_tls_cert_src_path: files/beegfs/cert/mgmtd_tls_cert.pem

beegfs_ha_tls_key_src_path: files/beegfs/cert/mgmtd_tls_key.pem

beegfs_ha_tls_config_options:
  alt_names: [<mgmt_service_ip>]
```

如果未提供 `beegfs_ha_tls_config_options.alt_names`，則 Ansible 將嘗試使用指定憑證/金鑰路徑中的現有憑證。

配置現有的 **BeeGFS v8** 叢集

對於現有的 BeeGFS 集群，您可以透過建立本機憑證授權單位並為管理服務產生必要的憑證來整合 TLS。更新 BeeGFS 管理服務設定檔中的路徑，使其指向新建立的憑證。

本節的說明僅供參考。處理私鑰和憑證時，應採取適當的安全預防措施。

建立憑證授權單位

在受信任的電腦上，建立一個本機憑證授權單位 (CA)，用於簽署 BeeGFS 管理服務的憑證。CA 憑證將分發給用戶端，以建立信任並實現與 BeeGFS 服務的安全通訊。

以下說明是在基於 RHEL 的系統上建立本機憑證授權單位的參考。

1. 如果尚未安裝 OpenSSL，請安裝它：

```
dnf install openssl
```

2. 建立用於儲存憑證檔案的工作目錄：

```
mkdir -p ~/beegfs_tls && cd ~/beegfs_tls
```

3. 產生 CA 私鑰：

```
openssl genrsa -out ca_key.pem 4096
```

4. 建立一個名為 `ca.cnf`的 CA 設定檔，並調整專有名稱欄位以符合您的組織：

```
[ req ]
default_bits        = 4096
distinguished_name  = req_distinguished_name
x509_extensions     = v3_ca
prompt              = no

[ req_distinguished_name ]
C  = <Country>
ST = <State>
L  = <City>
O  = <Organization>
OU = <OrganizationalUnit>
CN = BeeGFS-CA

[ v3_ca ]
basicConstraints = critical,CA:TRUE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer:always
```

5. 產生 CA 憑證。此憑證的有效期限應與系統生命週期相同，否則您需要在憑證過期前規劃重新產生憑證。憑證過期後，某些元件之間的通訊將無法進行，更新 TLS 憑證通常需要重新啟動服務才能完成。

   以下指令產生有效期限為 1 年的 CA 憑證：

```
openssl req -new -x509 -key ca_key.pem -out ca_cert.pem -days 365
-config ca.cnf
```

   ⓘ 雖然為了簡單起見，本範例使用了 1 年的有效期，但您應該根據貴組織的安全要求調整 `-days`參數，並建立憑證續約流程。

建立管理服務憑證

為您的 BeeGFS 管理服務產生證書，並使用您建立的 CA 對其進行簽署。這些證書將安裝在執行 BeeGFS 管理服務的檔案節點上。

1. 產生管理服務私鑰：

```
openssl genrsa -out mgmtd_tls_key.pem 4096
```

2. 建立一個名為 `tls_san.cnf` 的憑證設定檔，其中包含所有管理服務 IP 位址的主體別名（SAN）：

```
[ req ]
default_bits        = 4096
distinguished_name  = req_distinguished_name
req_extensions      = req_ext
prompt              = no

[ req_distinguished_name ]
C  = <Country>
ST = <State>
L  = <City>
O  = <Organization>
OU = <OrganizationalUnit>
CN = beegfs-mgmt

[ req_ext ]
subjectAltName = @alt_names

[ v3_ca ]
subjectAltName = @alt_names
basicConstraints = CA:FALSE

[ alt_names ]
IP.1 = <beegfs_mgmt_service_ip_1>
IP.2 = <beegfs_mgmt_service_ip_2>
```

更新專有名稱欄位以符合您的 CA 配置，並將 `IP.1`和 `IP.2`值更新為您的管理服務 IP 位址。

3. 產生憑證簽章請求 (CSR)：

```
openssl req -new -key mgmtd_tls_key.pem -out mgmtd_tls_csr.pem -config
tls_san.cnf
```

4. 使用您的 CA 簽署憑證（有效期為 1 年）：

```
openssl x509 -req -in mgmtd_tls_csr.pem -CA ca_cert.pem -CAkey
ca_key.pem -CAcreateserial -out mgmtd_tls_cert.pem -days 365 -sha256
-extensions v3_ca -extfile tls_san.cnf
```

根據貴組織的安全策略調整證書有效期限 (`-days 365`。許多組織要求每 1-2 年輪換一次證書。

5. 驗證憑證是否已正確建立：

```
openssl x509 -in mgmtd_tls_cert.pem -text -noout
```

請確認「主題備用名稱」部分包含所有管理 IP 位址。

將證書分發到檔案節點

將 CA 憑證和管理服務憑證分發到對應的檔案節點和用戶端。

1. 將 CA 憑證、管理服務憑證和金鑰複製到執行管理服務的檔案節點：

```
scp ca_cert.pem mgmtd_tls_cert.pem mgmtd_tls_key.pem
user@beegfs_01:/etc/beegfs/
scp ca_cert.pem mgmtd_tls_cert.pem mgmtd_tls_key.pem
user@beegfs_02:/etc/beegfs/
```

將管理服務指向 **TLS** 憑證

更新 BeeGFS 管理服務設定以啟用 TLS 並引用已建立的 TLS 憑證。

1. 在執行 BeeGFS 管理服務的檔案節點上，編輯管理服務設定檔，例如位於 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml`。新增或更新以下與 TLS 相關的參數：

```
tls-disable = false
tls-cert-file = "/etc/beegfs/mgmtd_tls_cert.pem"
tls-key-file = "/etc/beegfs/mgmtd_tls_key.pem"
```

2. 請採取適當措施，安全地重新啟動 BeeGFS 管理服務，以使變更生效：

```
systemctl restart beegfs-mgmtd
```

3. 驗證管理服務是否已成功啟動：

```
journalctl -xeu beegfs-mgmtd
```

查看日誌條目，確認 TLS 初始化和憑證載入是否成功。

```
Successfully initialized certificate verification library.
Successfully loaded license certificate: TMP-XXXXXXXXXX
```

為 **BeeGFS v8** 用戶端設定 **TLS**

建立並向所有需要與 BeeGFS 管理服務通訊的 BeeGFS 用戶端分發由本機 CA 簽署的憑證。

1. 使用與上述管理服務證書相同的流程為用戶端產生證書，但在 Subject Alternative Name (SAN) 欄位中使用用戶端的 IP 位址或主機名稱。
2. 將客戶端憑證安全地遠端複製到客戶端，並在客戶端上將該憑證重新命名為 `cert.pem`：

```
scp client_cert.pem user@client:/etc/beegfs/cert.pem
```

3. 在所有客戶端上重新啟動 BeeGFS 用戶端服務：

```
systemctl restart beegfs-client
```

4. 透過執行 `beegfs CLI` 命令驗證客戶端連線，例如：

```
beegfs health check
```

停用 **TLS**

TLS 可以停用，用於故障排除或使用者本身需要。但不建議這樣做，因為它會以明文形式暴露內部檔案系統結構和配置等潛在敏感資訊。請依照以下說明在現有或新建的 BeeGFS v8 叢集上停用 TLS。

部署新的 **BeeGFS v8** 叢集

對於新的 BeeGFS 叢集部署，可以透過在 Ansible 清單的 `user_defined_params.yml` 檔案中設定以下參數來停用 TLS 進行叢集部署：

```
beegfs_ha_tls_enabled: false
```

配置現有的 **BeeGFS v8** 叢集

對於現有的 BeeGFS v8 集群，請編輯管理服務設定檔。例如，編輯位於 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml`的檔案並設定：

```
tls-disable = true
```

採取適當措施安全地重新啟動管理服務，以使變更生效。

管理BeeGFS叢集

概述、主要概念和術語

瞭解如何在部署BeeGFS HA叢集之後管理這些叢集。

總覽

本節適用於部署BeeGFS HA叢集之後、需要管理的叢集管理員。即使是熟悉Linux HA叢集的使用者、也應該徹底閱讀本指南、因為管理叢集的方式有許多差異、尤其是使用Ansible進行重新設定。

重要概念

雖然在主"詞彙與概念"頁上介紹了其中一些概念、但在 BeeGFS HA 叢集的背景下重新介紹這些概念是很有幫助的：

**叢集節點：*執行起搏器和電暈器同步服務的伺服器、並參與HA叢集。

**檔案節點：*用於執行一或多個BeeGFS管理、中繼資料或儲存服務的叢集節點。

*區塊節點： NetApp E系列儲存系統、可為檔案節點提供區塊儲存。這些節點不會參與BeeGFS HA叢集、因為它們提供自己的獨立HA功能。每個節點包含兩個儲存控制器、可在區塊層提供高可用度。

- BeeGFS服務：* BeeGFS管理、中繼資料或儲存服務。每個檔案節點都會執行一或多項服務、這些服務會使用區塊節點上的磁碟區來儲存其資料。

**建置區塊：*標準化部署BeeGFS檔案節點、E系列區塊節點、以及其上執行的BeeGFS服務、可簡化BeeGFS HA叢集/檔案系統的擴充、並遵循NetApp驗證架構。也支援自訂HA叢集、但通常採用類似的建置區塊方法來簡化擴充。

- BeeGFS HA叢集：*一組可擴充的檔案節點、用於執行BeeGFS服務、並以區塊節點作為後盾、以高可用度的方式儲存BeeGFS資料。以業界公認的開放原始碼元件為基礎、採用Ansible進行包裝與部署。

**叢集服務：*是指叢集內每個節點上執行的起搏器和電暈器同步服務。請注意、如果只需要兩個檔案節點、節點可能無法執行任何BeeGFS服務、只能以「tiebreaker」節點的形式參與叢集。

**叢集資源：*針對叢集中執行的每個BeeGFS服務、您將會看到BeeGFS監控資源、以及一個資源群組、其中包含BeeGFS目標、IP位址（浮動IP）及BeeGFS服務本身的資源。

- Ansible：*軟體資源配置、組態管理及應用程式部署工具、以程式碼形式提供基礎架構。這是BeeGFS HA叢集的封裝方式、可簡化在NetApp上部署、重新設定及更新BeeGFS的程序。
- PCS：*叢集中任何檔案節點均提供命令列介面、用於查詢及控制叢集中節點和資源的狀態。

通用術語

**容錯移轉：*每個BeeGFS服務都有一個優先要執行的檔案節點、除非該節點故障。當BeeGFS服務在非慣用/次要檔案節點上執行時、就表示該服務正在進行容錯移轉。

*容錯回復：*將BeeGFS服務從非偏好的檔案節點移回偏好的節點。

- HA配對：*可存取相同區塊節點集的兩個檔案節點、有時稱為HA配對。這是整個NetApp常用的詞彙、用來指可「接管」彼此的兩個儲存控制器或節點。
- 維護模式： ** 停用所有資源監控"維護模式"功能、並防止 Pacemaker 移動或以其他方式管理叢集中的資源（請同時參閱上的一節）。
- HA叢集：*執行BeeGFS服務的一或多個檔案節點、可在叢集中的多個節點之間容錯移轉、以建立高可用度的BeeGFS檔案系統。檔案節點通常會設定成HA配對、以便在叢集中執行BeeGFS服務的子集。

使用Ansible與PCS工具的時機

您應該何時使用Ansible與PCS命令列工具來管理HA叢集？

所有叢集部署和重新設定工作都應使用外部可控制節點的Ansible來完成。叢集狀態的暫時性變更（例如、將節點置入和移出待命）通常是透過登入叢集的一個節點（最好是未降級或即將進行維護的節點）、以及使用PCS命令列工具來執行。

修改任何叢集組態、包括資源、限制、內容及BeeGFS服務本身、都應該使用Ansible來完成。維護叢集的一部分、是維護「Ansible Inventory and playbook（Ansible Inventory and playbook）」的最新複本（理想的來源控制方式是追蹤變更）。當您需要變更組態時、請更新詳細目錄、然後重新執行匯入BeeGFS HA角色的Ansible教戰手冊。

HA角色會處理將叢集置於維護模式、然後在重新啟動BeeGFS或叢集服務以套用新組態之前進行任何必要的變更。由於在初始部署之外通常不需要完整的節點重新開機、因此重新執行Ansible通常被視為「安全」程序、但如果任何BeeGFS服務需要重新啟動、則建議在維護期間或下班時間重新執行。這些重新啟動通常不會造成應用程式錯誤、但可能會影響效能（某些應用程式處理效能可能優於其他應用程式）。

當您想要將整個叢集恢復到完全最佳狀態時、重新執行Ansible也是一個選項、而且在某些情況下、可能比使用PC更容易恢復叢集狀態。尤其是在叢集因某種原因而停機的緊急情況下、一旦所有節點都備份重新執行Ansible、就可能比嘗試使用PC更快且更可靠地還原叢集。

檢查叢集的狀態

使用PC檢視叢集的狀態。

總覽

執行中 `pcs status` 從任何叢集節點、都是查看叢集整體狀態及每個資源狀態（例如BeeGFS服務及其相依性）的最簡單方法。本節將說明您在的輸出中所能找到的內容 `pcs status` 命令。

瞭解輸出來源 `pcs status`

執行 `pcs status` 在任何叢集服務（起搏器和電暈器同步）啟動的叢集節點上。輸出頂端會顯示叢集摘要：

```
[root@beegfs_01 ~]# pcs status
Cluster name: hacluster
Cluster Summary:
  * Stack: corosync
  * Current DC: beegfs_01 (version 2.0.5-9.el8_4.3-ba59be7122) - partition
with quorum
  * Last updated: Fri Jul  1 13:37:18 2022
  * Last change:  Fri Jul  1 13:23:34 2022 by root via cibadmin on
beegfs_01
  * 6 nodes configured
  * 235 resource instances configured
```

以下章節列出叢集中的節點：

```
Node List:
  * Node beegfs_06: standby
  * Online: [ beegfs_01 beegfs_02 beegfs_04 beegfs_05 ]
  * OFFLINE: [ beegfs_03 ]
```

這特別代表任何處於待命或離線狀態的節點。待命中的節點仍在參與叢集、但標記為不符合執行資源的資格。離線的節點表示叢集服務未在該節點上執行、可能是因為手動停止、或是因為節點已重新開機/關機。

當節點初次啟動時、叢集服務將會停止、需要手動啟動、以避免意外將資源還原至不正常的節點。

如果節點因為非管理原因（例如故障）而處於待命或離線狀態、則會在節點狀態旁以括弧顯示其他文字。例如、如果隔離功能已停用、而資源遇到您將會看到的故障 `Node <HOSTNAME>: standby (on-fail)`。另一個可能的狀態是 `Node <HOSTNAME>: UNCLEAN (offline)`（這會短暫地被視為節點正在被圍起來、但如果隔離失敗、表示叢集無法確認節點狀態、則會持續存在（這可能會封鎖其他節點上的資源啟動）。

下一節顯示叢集中所有資源及其狀態的清單：

```
Full List of Resources:
  * mgmt-monitor    (ocf::eseries:beegfs-monitor):   Started beegfs_01
  * Resource Group: mgmt-group:
    * mgmt-FS1  (ocf::eseries:beegfs-target):    Started beegfs_01
    * mgmt-IP1  (ocf::eseries:beegfs-ipaddr2):   Started beegfs_01
    * mgmt-IP2  (ocf::eseries:beegfs-ipaddr2):   Started beegfs_01
    * mgmt-service  (systemd:beegfs-mgmtd):  Started beegfs_01
[...]
```

與節點類似、如果資源有任何問題、資源狀態旁會顯示其他文字、並以括弧表示。例如、如果心臟起搏器要求資源停止、但無法在分配的時間內完成、則心臟起搏器會嘗試隔離節點。如果禁用隔離功能或隔離操作失敗，則資源狀態將是 `FAILED <HOSTNAME> (blocked)` 而起搏器將無法在不同的節點上啟動。

值得一提的是、BeeGFS HA叢集運用了許多BeeGFS最佳化的自訂OCF資源代理程式。特別是BeeGFS監視器、負責在特定節點上的BeeGFS資源無法使用時觸發容錯移轉。

重新設定HA叢集和BeeGFS

使用Ansible重新設定叢集。

總覽

一般而言、只要更新 Ansible 清查並重新執行 `ansible-playbook`命令、即可重新設定 BeeGFS HA 叢集的任何層面。這包括更新警示、變更永久隔離組態或調整BeeGFS服務組態。您 `group_vars/ha_cluster.yml`可以使用檔案來調整這些選項、並在"指定通用檔案節點組態"一節中找到完整的選項清單。

如需管理員在執行維護或服務叢集時應注意的特定組態選項詳細資料、請參閱下方。

如何停用和啟用屏障

設定叢集時、預設會啟用/需要隔離功能。在某些情況下、可能需要暫時停用隔離功能、以確保在執行某些維護作業（例如升級作業系統）時、不會意外關閉節點。雖然可以手動停用此功能、但系統管理員仍應注意取捨。

選項**1**：使用**Ansible**（建議）停用隔離功能。

使用Ansible停用隔離功能時、BeeGFS監視器的失敗動作會從「Fence」變更為「standby」（待命）。這表示、如果BeeGFS監視器偵測到故障、就會嘗試將節點置於待命狀態、並容錯移轉所有BeeGFS服務。在主動式疑難排解/測試之外、這通常比選項2更為理想。缺點是、如果資源無法在原始節點上停止、就會被封鎖、無法從其他位置啟動（這也是為什麼正式作業叢集通常需要隔離）。

1. 在您的Ansible庫存中 `groups_vars/ha_cluster.yml` 新增下列組態：

```
beegfs_ha_cluster_crm_config_options:
    stonith-enabled: False
```

2. 重新執行「Ansible」方針、將變更套用至叢集。

選項**2**：手動停用隔離功能。

在某些情況下、您可能想要暫時停用隔離功能、而不重新執行Ansible、或許是為了協助疑難排解或測試叢集。

在此組態中、如果BeeGFS監視器偵測到故障、叢集將嘗試停止對應的資源群組。它不會觸發完整容錯移轉、也不會嘗試重新啟動受影響的資源群組、或將其移至其他主機。若要恢復、請先解決所有問題、然後再執行 `pcs resource cleanup` 或手動將節點置於待命狀態。

步驟：

1. 若要判斷隔離（stonith）是否已全域啟用或停用、請執行： `pcs property show stonith-enabled`
2. 若要停用隔離執行： `pcs property set stonith-enabled=false`
3. 若要啟用隔離執行： `pcs property set stonith-enabled=true`

下次執行 Ansible 劇本時，此設定將被覆蓋。

更新 HA 叢集元件

升級 BeeGFS 服務

使用 Ansible 更新 HA 叢集上執行的 BeeGFS 版本。

總覽

BeeGFS 遵循 `major.minor.patch`版本管理方案。`BeeGFS HA Ansible` 角色適用於每個支援的 `major.minor`版本（例如、 `beegfs_ha_7_2`和 `beegfs_ha_7_3`）。每個 HA 角色都固定在 Ansible 集合發行時可用的最新 BeeGFS 修補程式版本上。

所有 BeeGFS 升級，包括 BeeGFS 主版本、次版本和補丁版本之間的遷移，都應該使用 Ansible。要更新 BeeGFS，您首先需要更新 BeeGFS Ansible 集合，這也會引入部署/管理自動化和底層高可用性叢集的最新修復和增強功能。即使更新到最新版本的集合，在執行 `ansible-playbook`並設定 `-e "beegfs_ha_force_upgrade=true"`之前，BeeGFS 不會升級。有關每次升級的更多詳細資訊，請參閱"BeeGFS 升級文件"您當前版本的文件。

如果您要升級到 BeeGFS v8，請參閱 "升級至 BeeGFS v8" 相關步驟。

已測試的升級途徑

以下升級路徑已通過測試和驗證：

原始版本	升級版本	多重軌道	詳細資料
7.2.6	7.3.2	是的	將beegfs集合從v3.0.1升級至v3.1.0、新增多重軌道
7.2.6	7.2.8	否	將beegfs集合從v3.0.1升級至v3.1.0
7.2.8	7.3.1	是的	使用beegfs集合v3.1.0升級、新增多重軌道
7.3.1	7.3.2	是的	使用beegfs集合v3.1.0升級
7.3.2	7.4.1..	是的	使用beegfs集合v3.2.0升級
7.4.1..	7.4.2..	是的	使用beegfs集合v3.2.0升級
7.4.2..	7.4.6	是的	使用beegfs集合v3.2.0升級
7.4.6	8.0	是的	請按照 "升級至 BeeGFS v8" 操作步驟中的說明進行升級。
7.4.6	8.1	是的	請按照 "升級至 BeeGFS v8" 操作步驟中的說明進行升級。
7.4.6	8.2	是的	請按照 "升級至 BeeGFS v8" 操作步驟中的說明進行升級。

BeeGFS 升級步驟

下列各節提供更新 BeeGFS Ansible 系列和 BeeGFS 本身的步驟。請特別注意任何額外步驟、以更新BeeGFS主要或次要版本。

步驟 1 ：升級 BeeGFS 系列

可存取的集合升級 "Ansible Galaxy"，執行下列命令：

```
ansible-galaxy collection install netapp_eseries.beegfs --upgrade
```

如需離線收藏升級、請從下載收藏 "Ansible Galaxy" 按一下所需的 `Install Version`、然後 `Download tarball`。將tar傳輸到Ansible控制節點、然後執行下列命令。

```
ansible-galaxy collection install netapp_eseries-beegfs-<VERSION>.tar.gz
--upgrade
```

請參閱 "安裝集合" 以取得更多資訊。

步驟 2 ：更新 Ansible 庫存

對叢集的 Ansible 清單檔案進行任何必要或所需的更新。有關具體升級要求的詳細資訊，請參閱以下 [版本升級注意事項] 部分。有關配置 BeeGFS HA 清單的一般資訊，請參閱 "Ansible Inventory Overview" 部分。

步驟 3 ：更新 Ansible 教戰手冊（僅更新主要或次要版本時）

如果您要在主要或次要版本之間移動、請在 `playbook.yml` 用於部署和維護叢集的檔案中、更新角色名稱 `beegfs_ha_<VERSION>` 以反映所需的版本。例如，如果您想部署 BeeGFS 7.4 ，這將是 `beegfs_ha_7_4`：

```
- hosts: all
  gather_facts: false
  any_errors_fatal: true
  collections:
    - netapp_eseries.beegfs
  tasks:
    - name: Ensure BeeGFS HA cluster is setup.
      ansible.builtin.import_role:  # import_role is required for tag
availability.
        name: beegfs_ha_7_4
```

如需此教戰手冊檔案內容的詳細資訊"部署BeeGFS HA叢集"、請參閱一節。

步驟 4 ：執行 BeeGFS 升級

若要套用BeeGFS更新：

```
ansible-playbook -i inventory.yml beegfs_ha_playbook.yml -e
"beegfs_ha_force_upgrade=true" --tags beegfs_ha
```

BeeGFS HA角色將在幕後處理：

- 確保叢集處於最佳狀態、且每個BeeGFS服務都位於其偏好的節點上。
- 將叢集置於維護模式。
- 更新HA叢集元件（如有需要）。
- 每次升級一個檔案節點、如下所示：
  - 將其置於待命狀態、並將其服務容錯移轉至次要節點。
  - 升級BeeGFS套件。
  - 回復服務：
- 將叢集移出維護模式。

版本升級注意事項

從**BeeGFS 7.2.6**或**7.3.0**版升級

連線型驗證變更

BeeGFS 版本 7.3.2 及更高版本需要設定基於連線的身份驗證。若缺少以下任一項，服務將無法啟動：

- 指定一個 `connAuthFile`，或
- 在服務的設定檔中進行設定 `connDisableAuthentication=true`。

強烈建議啟用基於連線的身份驗證以確保安全。請參閱 "BeeGFS連線型驗證" 以獲取更多資訊。

這些 `beegfs_ha*` 角色會自動產生身份驗證檔案並將其分發給：

- 叢集中的所有檔案節點
- Ansible 控制節點位於 `<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile`

該 `beegfs_client`角色會在此檔案存在時自動偵測並將其套用到用戶端。

> 如果您未使用 `beegfs_client` 角色設定用戶端，則必須手動將身份驗證檔案分發給每個用戶端，並在 `beegfs-client.conf` 檔案中設定 `connAuthFile` 設定。從不支援基於連線的身份驗證的 BeeGFS 版本升級時，除非您在升級過程中透過在 `group_vars/ha_cluster.yml` 中設定 `beegfs_ha_conn_auth_enabled: false` 來停用基於連線的身份驗證（不建議這樣做），否則用戶端將失去存取權限。

如需更多詳細資訊和替代設定選項，請參閱 "指定通用檔案節點組態" 部分中的連線驗證設定步驟。

升級至 **BeeGFS v8**

請依照以下步驟將您的 BeeGFS HA 叢集從版本 7.4.6 升級到 BeeGFS v8。

總覽

BeeGFS v8 引入了多項重大變更，從 BeeGFS v7 升級之前需要額外的設定。本文檔將指導您如何為 BeeGFS v8 的新要求準備叢集，然後升級到 BeeGFS v8。

在升級到 BeeGFS v8 之前，請確保您的系統至少運行 BeeGFS 7.4.6。任何執行 BeeGFS 7.4.6 之前版本的叢集都必須先"升級至 7.4.6 版本"，然後再繼續執行此 BeeGFS v8 升級程序。

BeeGFS v8 的主要變化

BeeGFS v8 引入了以下主要變化：

- 授權強制執行： BeeGFS v8 需要授權才能使用進階功能，例如儲存池、遠端儲存目標、BeeOND 等。在升級之前，請為您的 BeeGFS 叢集取得有效的授權。如有需要，您可以從 "BeeGFS 授權入口網站" 取得臨時 BeeGFS v8 評估授權。
- 管理服務資料庫遷移： 若要啟用 BeeGFS v8 中基於 TOML 的新格式的配置，您必須手動將 BeeGFS v7 管理服務資料庫遷移到更新的 BeeGFS v8 格式。
- **TLS** 加密： BeeGFS v8 引入了 TLS 加密，用於服務間的安全通訊。升級過程中，您需要為 BeeGFS 管理服務和 `beegfs` 命令列公用程式產生並分發 TLS 憑證。

有關 BeeGFS 8 的更多詳細資訊和其他更改，請參閱 "BeeGFS v8.0.0 升級指南"。

升級到 BeeGFS v8 需要叢集停機維護。此外，BeeGFS v7 用戶端無法連線到 BeeGFS v8 叢集。請仔細協調叢集和用戶端的升級時間，以最大程度地減少對運作的影響。

準備好 **BeeGFS** 叢集以進行升級

在開始升級之前，請仔細準備您的環境，以確保平穩過渡並最大限度地減少停機時間。

1. 確保叢集處於健康狀態,所有 BeeGFS 服務都在其首選節點上運行。從執行 BeeGFS 服務的檔案節點,驗證所有 Pacemaker 資源是否都在其首選節點上執行:

```
pcs status
```

2. 記錄並備份您的叢集組態。
   a. 請參閱 "BeeGFS 備份文檔" 以取得備份叢集組態的相關說明。
   b. 備份現有管理資料目錄：

```
cp -r /mnt/mgmt_tgt_mgmt01/data
/mnt/mgmt_tgt_mgmt01/data_beegfs_v7_backup_$(date +%Y%m%d)
```

   c. 從 beegfs 用戶端執行以下命令，並儲存其輸出以供參考：

```
beegfs-ctl --getentryinfo --verbose /path/to/beegfs/mountpoint
```

   d. 如果使用鏡像，請收集詳細的狀態資訊：

```
beegfs-ctl --listtargets --longnodes --state --spaceinfo
--mirrorgroups --nodetype=meta
beegfs-ctl --listtargets --longnodes --state --spaceinfo
--mirrorgroups --nodetype=storage
```

3. 讓客戶做好服務中斷的準備並停止 `beegfs-client`服務。對每個客戶執行：

```
systemctl stop beegfs-client
```

4. 對於每個 Pacemaker 叢集，停用 STONITH。這樣可以在升級後驗證叢集的完整性，而不會觸發不必要的節點重新啟動。

```
pcs property set stonith-enabled=false
```

5. 對於 BeeGFS 命名空間中的所有 Pacemaker 叢集，請使用 PCS 停止叢集：

```
pcs cluster stop --all
```

升級 **BeeGFS** 套件

在叢集中的所有檔案節點上，新增適用於您 Linux 發行版的 BeeGFS v8 軟體包倉庫。有關使用官方 BeeGFS 倉庫的說明，請造訪 "BeeGFS 下載頁面"。或者，請相應地配置您的本地 BeeGFS 鏡像倉庫。

以下步驟示範如何在 RHEL 9 檔案節點上使用官方 BeeGFS 8.2 儲存庫。請在叢集中的所有檔案節點上執行下列步驟：

1. 匯入 BeeGFS GPG 金鑰：

```
rpm --import https://www.beegfs.io/release/beegfs_8.2/gpg/GPG-KEY-beegfs
```

2. 匯入 BeeGFS 儲存庫：

```
curl -L -o /etc/yum.repos.d/beegfs-rhel9.repo
https://www.beegfs.io/release/beegfs_8.2/dists/beegfs-rhel9.repo
```

> ⓘ 刪除先前配置的所有 BeeGFS 儲存庫，以避免與新的 BeeGFS v8 儲存庫發生衝突。

3. 清除套件管理器快取：

```
dnf clean all
```

4. 在所有檔案節點上，將 BeeGFS 套件更新至 BeeGFS 8.2。

```
dnf update beegfs-mgmtd beegfs-storage beegfs-meta libbeegfs-ib
```

在標準叢集中， `beegfs-mgmtd` 軟體包只會更新前兩個檔案節點。

升級管理資料庫

在執行 BeeGFS 管理服務的檔案節點之一上，執行下列步驟將管理資料庫從 BeeGFS v7 遷移到 v8。

1. 列出所有 NVMe 裝置並篩選管理目標：

```
nvme netapp smdevices | grep mgmt_tgt
```

a. 請注意輸出中的裝置路徑。

b. 將管理目標裝置掛載到現有的管理目標掛載點（將 `/dev/nvmeXnY` 替換為您的裝置路徑）：

```
mount /dev/nvmeXnY /mnt/mgmt_tgt_mgmt01/
```

2. 執行以下命令，將您的 BeeGFS 7 管理資料匯入新的資料庫格式：

```
/opt/beegfs/sbin/beegfs-mgmtd --import-from
-v7=/mnt/mgmt_tgt_mgmt01/data/
```

預期輸出：

```
Created new database version 3 at "/var/lib/beegfs/mgmtd.sqlite".
Successfully imported v7 management data from
"/mnt/mgmt_tgt_mgmt01/data/".
```

由於 BeeGFS v8 中更嚴格的驗證要求，自動導入可能無法在所有情況下成功。例如，如果目標被指派到不存在的儲存池，則導入將失敗。若遷移失敗，請勿繼續升級。請聯絡 NetApp 支援以取得資料庫遷移問題的協助。作為臨時解決方案，您可以降級 BeeGFS v8 軟體包，並在問題解決期間繼續運行 BeeGFS v7。

3. 將產生的 SQLite 檔案移到管理服務掛載點：

```
mv /var/lib/beegfs/mgmtd.sqlite /mnt/mgmt_tgt_mgmt01/data/
```

4. 將產生的 `beegfs-mgmtd.toml`移至管理服務掛載點：

```
mv /etc/beegfs/beegfs-mgmtd.toml /mnt/mgmt_tgt_mgmt01/mgmt_config/
```

準備 `beegfs-mgmtd.toml`設定檔將在完成下一節中的授權和 TLS 設定步驟之後進行。

設定授權

1. 在所有執行 beegfs 管理服務的節點上安裝 beegfs 授權套件。通常情況下，這是叢集的前兩個節點：

```
dnf install libbeegfs-license
```

2. 將您的 BeeGFS v8 授權檔案下載到管理節點並放置在：

```
/etc/beegfs/license.pem
```

設定 **TLS** 加密

BeeGFS v8 要求管理服務和用戶端之間使用 TLS 加密以確保通訊安全。配置管理服務和用戶端服務之間網路通訊的 TLS 加密有三種方法。推薦且最安全的方法是使用受信任的憑證授權單位（CA）簽署的憑證。或者，您可以建立自己的本機 CA 來為 BeeGFS 叢集簽署憑證。對於不需要加密的環境或故障排除，可以完全停用 TLS，但不建議這樣做，因為它會將敏感資訊暴露在網路中。

在繼續操作之前，請按照 "為 BeeGFS 8 設定 TLS 加密" 指南中的說明為您的環境設定 TLS 加密。

更新管理服務配置

將設定從 BeeGFS v7 設定檔手動轉移到 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml`檔案中，以準備 BeeGFS v8 管理服務設定檔。

1. 在已掛載管理目標的管理節點上，引用 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.conf` BeeGFS 7 的管理服務文件，然後將所有設定複製到該 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` 文件中。對於基本設置，您的 `beegfs-mgmtd.toml` 配置可能如下所示：

```
beemsg-port = 8008
grpc-port = 8010
log-level = "info"
node-offline-timeout = "900s"
quota-enable = false
auth-disable = false
auth-file = "/etc/beegfs/<mgmt_service_ip>_connAuthFile"
db-file = "/mnt/mgmt_tgt_mgmt01/data/mgmtd.sqlite"
license-disable = false
license-cert-file = "/etc/beegfs/license.pem"
tls-disable = false
tls-cert-file = "/etc/beegfs/mgmtd_tls_cert.pem"
tls-key-file = "/etc/beegfs/mgmtd_tls_key.pem"
interfaces = ['i1b:mgmt_1', 'i2b:mgmt_2']
```

根據您的環境和 TLS 配置調整所有路徑。

2. 在每個執行管理服務的檔案節點上，修改 systemd 服務檔案，使其指向新的組態檔位置。

```
sudo sed -i 's|ExecStart=.*|ExecStart=nice -n -3
/opt/beegfs/sbin/beegfs-mgmtd --config-file
/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml|'
/etc/systemd/system/beegfs-mgmtd.service
```

   a. 重新載入 systemd：

```
systemctl daemon-reload
```

3. 對於執行管理服務的每個檔案節點，開啟連接埠 8010 以進行管理服務的 gRPC 通訊。

   a. 將連接埠 8010/tcp 新增至 beegfs 區域：

```
sudo firewall-cmd --zone=beegfs --permanent --add-port=8010/tcp
```

   b. 重新載入防火牆以套用變更：

```
sudo firewall-cmd --reload
```

更新 **BeeGFS** 監控腳本

Pacemaker `beegfs-monitor` OCF 腳本需要更新以支援新的 TOML 設定格式和 systemd 服務管理。請在叢集中的一個節點上更新腳本，然後將更新後的腳本複製到所有其他節點。

1. 建立目前腳本的備份：

```
cp /usr/lib/ocf/resource.d/eseries/beegfs-monitor
/usr/lib/ocf/resource.d/eseries/beegfs-monitor.bak.$(date +%F)
```

2. 將管理設定檔路徑從 `.conf` 更新為 `.toml`：

```
sed -i 's|mgmt_config/beegfs-mgmtd\.conf|mgmt_config/beegfs-mgmtd.toml|'
/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

或者，手動在腳本中找到以下程式碼區塊：

```
case $type in
  management)
    conf_path="${configuration_mount}/mgmt_config/beegfs-mgmtd.conf"
    ;;
```

並將其替換為：

```
case $type in
  management)
    conf_path="${configuration_mount}/mgmt_config/beegfs-mgmtd.toml"
    ;;
```

3. 更新 `get_interfaces()` 和 `get_subnet_ips()` 函數以支援 TOML 配置：
   a. 在文字編輯器中開啟腳本：

```
vi /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

   b. 找出這兩個函數：`get_interfaces()` 和 `get_subnet_ips()`。
   c. 刪除這兩個完整函數，從 `get_interfaces()` 開始到 `get_subnet_ips()` 結尾。
   d. 請將以下更新的函數複製並貼上到對應位置：

```
# Return network communication interface name(s) from the BeeGFS
resource's connInterfaceFile
get_interfaces() {
  # Determine BeeGFS service network IP interfaces.
  if [ "$type" = "management" ]; then
    interfaces_line=$(grep "^interfaces =" "$conf_path")
    interfaces_list=$(echo "$interfaces_line" | sed "s/.*= \[\(.*
\)\]/\1/")
    interfaces=$(echo "$interfaces_list" | tr -d "'" | tr -d " " | tr
',' '\n')

    for entry in $interfaces; do
      echo "$entry" | cut -d ':' -f 1
    done
  else
    connInterfacesFile_path=$(grep "^connInterfacesFile" "$conf_path"
| tr -d "[:space:]" | cut -f 2 -d "=")

    if [ -f "$connInterfacesFile_path" ]; then
      while read -r entry; do
        echo "$entry" | cut -f 1 -d ':'
      done < "$connInterfacesFile_path"
    fi
  fi
}

# Return list containing all the BeeGFS resource's usable IP
addresses. *Note that these are filtered by the connNetFilterFile
entries.
get_subnet_ips() {
  # Determine all possible BeeGFS service network IP addresses.
  if [ "$type" != "management" ]; then
    connNetFilterFile_path=$(grep "^connNetFilterFile" "$conf_path" |
tr -d "[:space:]" | cut -f 2 -d "=")

    filter_ips=""
    if [ -n "$connNetFilterFile_path" ] && [ -e
$connNetFilterFile_path ]; then
      while read -r filter; do
        filter_ips="$filter_ips $(get_ipv4_subnet_addresses $filter)"
      done < $connNetFilterFile_path
    fi

    echo "$filter_ips"
  fi
}
```

e. 儲存並退出文字編輯器。

f. 執行以下命令以在繼續之前檢查腳本的語法錯誤。沒有輸出表示腳本語法正確。

```
bash -n /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

4. 將更新後的 `beegfs-monitor` OCF 腳本複製到叢集中的所有其他節點，以確保一致性：

```
scp /usr/lib/ocf/resource.d/eseries/beegfs-monitor
user@node:/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

使叢集恢復上線

1. 完成之前的所有升級步驟後，透過在所有節點上啟動 BeeGFS 服務，使叢集重新上線。

```
pcs cluster start --all
```

2. 驗證 `beegfs-mgmtd` 服務是否成功啟動：

```
journalctl -xeu beegfs-mgmtd
```

預期輸出包含如下行：

```
Started Cluster Controlled beegfs-mgmtd.
Loaded config file from "/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-
mgmtd.toml"
Successfully initialized certificate verification library.
Successfully loaded license certificate: TMP-113489268
Opened database at "/mnt/mgmt_tgt_mgmt01/data/mgmtd.sqlite"
Listening for BeeGFS connections on [::]:8008
Serving gRPC requests on [::]:8010
```

如果日誌中出現錯誤，請檢查管理設定檔路徑，並確保所有值已從 BeeGFS 7 設定檔正確傳輸。

3. 執行 `pcs status` 並驗證叢集是否運作正常，以及各項服務是否已在首選節點上啟動。

4. 確認叢集運作狀況正常後，重新啟用 STONITH：

```
pcs property set stonith-enabled=true
```

5. 接下來進入下一節，升級叢集中的 BeeGFS 用戶端，並檢查 BeeGFS 叢集的運作狀況。

升級 **BeeGFS** 用戶端

成功將叢集升級至 BeeGFS v8 後，您還必須升級所有 BeeGFS 用戶端。

以下步驟概述了在基於 Ubuntu 的系統上升級 BeeGFS 用戶端的過程。

1. 如果尚未停止，請停止 BeeGFS 用戶端服務：

```
systemctl stop beegfs-client
```

2. 為您的 Linux 發行版新增 BeeGFS v8 軟體包倉庫。有關使用官方 BeeGFS 倉庫的說明，請造訪"^BeeGFS 下載頁面"。或者，請相應地配置您的本地 BeeGFS 鏡像倉庫。

   以下步驟在基於 Ubuntu 的系統上使用官方 BeeGFS 8.2 軟體倉庫：

3. 匯入 BeeGFS GPG 金鑰：

```
wget https://www.beegfs.io/release/beegfs_8.2/gpg/GPG-KEY-beegfs -O
/etc/apt/trusted.gpg.d/beegfs.asc
```

4. 下載儲存庫檔案：

```
wget https://www.beegfs.io/release/beegfs_8.2/dists/beegfs-noble.list -O
/etc/apt/sources.list.d/beegfs.list
```

   ⓘ 刪除先前配置的所有 BeeGFS 儲存庫，以避免與新的 BeeGFS v8 儲存庫發生衝突。

5. 升級 BeeGFS 用戶端套件：

```
apt-get update
apt-get install --only-upgrade beegfs-client
```

6. 為客戶端配置 TLS。使用 BeeGFS CLI 需要 TLS。請參考 "為 BeeGFS 8 設定 TLS 加密" 步驟在客戶端設定 TLS。
7. 啟動 BeeGFS 用戶端服務：

```
systemctl start beegfs-client
```

驗證升級

完成升級至 BeeGFS v8 後，執行以下命令以驗證升級是否成功。

1. 驗證根 inode 是否仍由先前的相同元資料節點擁有。如果您使用了管理服務中的 `import-from-v7` 功能，則此過程應該會自動完成：

```
beegfs entry info /mnt/beegfs
```

2. 確認所有節點和目標均在線上且狀態良好：

```
beegfs health check
```

如果「可用容量」檢查警告目標可用空間不足，您可以調整 `beegfs-mgmtd.toml` 檔案中定義的「容量池」閾值，使其更適合您的環境。

升級 **HA** 叢集中的 **Pacemaker** 和 **corosync** 套件

請依照下列步驟升級 HA 叢集中的節律器和電暈器同步套件。

總覽

升級 Pacemaker 和 corosync 可確保叢集從新功能，安全修補程式和效能改善中獲益。

升級方法

升級叢集有兩種建議方法：滾動升級或完全關閉叢集。每種方法都有自己的優缺點。您的升級程序可能會因心律調整器版本而異。請參閱 ClusterLabs 的"升級 Pacemaker 叢集"說明文件，以判斷要使用哪種方法。在遵循升級方法之前，請確認：

- NetApp BeeGFS 解決方案支援全新的節律器和電暈器同步套件。
- BeeGFS 檔案系統和 Pacemaker 叢集組態有有效的備份。
- 叢集處於正常狀態。

循環升級

此方法涉及從叢集中移除每個節點，將其升級，然後將其重新引入叢集，直到所有節點都執行新版本為止。這種方法可讓叢集持續運作，這是較大型 HA 叢集的理想選擇，但在處理過程中會有執行混合版本的風險。在雙節點叢集中，應避免使用此方法。

1. 確認叢集處於最佳狀態，且每個 BeeGFS 服務都在其偏好的節點上執行。如 "檢查叢集的狀態" 需詳細資訊、請參閱。
2. 若要升級節點，請將其置入待命模式，以耗盡（或移動）所有 BeeGFS 服務：

```
pcs node standby <HOSTNAME>
```

3. 執行下列步驟，確認節點的服務已耗盡：

```
pcs status
```

請確定未將任何服務回報為待命節點上的服務 `Started`。

視叢集大小而定，服務可能需要幾秒鐘或幾分鐘才能移至姊妹節點。如果 BeeGFS 服務無法在姊妹節點上啟動"疑難排解指南"，請參閱。

4. 關閉節點上的叢集：

```
pcs cluster stop <HOSTNAME>
```

5. 升級節點上的 Pacemaker ， corossync 和 PCS 套件：

套件管理員命令會因作業系統而異。下列命令適用於執行 RHEL 8 及後續版本的系統。

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

6. 在節點上啟動 Pacemaker 叢集服務：

```
pcs cluster start <HOSTNAME>
```

7. 如果 `pcs`套件已更新，請使用叢集重新驗證節點：

```
pcs host auth <HOSTNAME>
```

8. 確認此工具的節律器組態仍然有效 `crm_verify`。

只需在叢集升級期間驗證一次即可。

```
crm_verify -L -V
```

9. 將節點從待機狀態移出：

```
pcs node unstandby <HOSTNAME>
```

10. 將所有 BeeGFS 服務重新部署回其偏好的節點：

```
pcs resource relocate run
```

11. 針對叢集中的每個節點重複上述步驟，直到所有節點都執行所需的節律器，電暈器同步和 PCS 版本為止。
12. 最後，請執行 `pcs status`並確認叢集狀況良好，並 `Current DC`回報所需的節律器版本。

> 如果 `Current DC`報告為「 ixed 版本」，則叢集中的某個節點仍在使用舊版 Pacemaker 執行，需要升級。如果任何升級的節點無法重新加入叢集，或資源無法啟動，請查看叢集記錄，並參閱 Pacemaker 版本說明或使用者指南，以瞭解已知的升級問題。

完成叢集關機

在此方法中，所有叢集節點和資源都會關閉，節點會升級，然後重新啟動叢集。如果節律器和電暈器同步版本不支援混合版本的組態，則必須使用此方法。

1. 確認叢集處於最佳狀態，且每個 BeeGFS 服務都在其偏好的節點上執行。如 "檢查叢集的狀態" 需詳細資訊、請參閱。
2. 關閉所有節點上的叢集軟體（ Pacemaker 和 corosync ）。

> 視叢集大小而定，整個叢集可能需要幾秒鐘或幾分鐘才能停止。

```
pcs cluster stop --all
```

3. 當所有節點上的叢集服務都關閉之後，請根據您的需求，升級每個節點上的 Pacemaker ， corosync 和 PCS 套件。

> 套件管理員命令會因作業系統而異。下列命令適用於執行 RHEL 8 及後續版本的系統。

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

4. 升級所有節點之後，請在所有節點上啟動叢集軟體：

```
pcs cluster start --all
```

5. 如果 `pcs`套件已更新，請重新驗證叢集中的每個節點：

```
pcs host auth <HOSTNAME>
```

6. 最後，請執行 `pcs status`並確認叢集狀況良好，並 `Current DC`報告正確的節律器版本。

如果 `Current DC`報告為「 ixed 版本」，則叢集中的某個節點仍在使用舊版 Pacemaker 執行，需要升級。

更新檔案節點介面卡韌體

請依照下列步驟，將檔案節點的 ConnectX-7 介面卡更新為最新的韌體。

總覽

可能需要更新 ConnectX-7 介面卡韌體，才能支援新的 MLNX_OFED 驅動程式，啟用新功能或修正錯誤。本指南將使用 NVIDIA 的 `mlxfwmanager`公用程式進行介面卡更新，因為它易於使用且效率高。

升級考量

本指南涵蓋兩種更新 ConnectX-7 介面卡韌體的方法：滾動更新和雙節點叢集更新。根據叢集的大小，選擇適當的更新方法。執行韌體更新之前，請確認：

- 已安裝支援的 MLNX_OFED 驅動程式，請參閱"技術需求"。
- BeeGFS 檔案系統和 Pacemaker 叢集組態有有效的備份。
- 叢集處於正常狀態。

韌體更新準備

建議您使用 NVIDIA `mlxfwmanager` 公用程式來更新節點的介面卡韌體，此韌體與 NVIDIA 的 MLNx_OFED 驅動程式一起提供。開始更新之前，請先從下載介面卡的韌體映像"NVIDIA 的支援網站"，並將其儲存在每個檔案節點上。

對於 Lenovo ConnectX-7 介面卡，請使用 `mlxfwmanager_LES` NVIDIA 頁面上的工具"OEM 韌體"。

滾動更新方法

建議任何具有兩個以上節點的 HA 叢集使用此方法。這種方法涉及一次在一個檔案節點上更新介面卡韌體，讓 HA 叢集能夠保留服務要求，不過建議您在此期間避免服務 I/O 。

1. 確認叢集處於最佳狀態，且每個 BeeGFS 服務都在其偏好的節點上執行。如 "檢查叢集的狀態" 需詳細資訊、請參閱。
2. 選擇要更新的檔案節點，並將其置於待命模式，以從該節點移除（或移動）所有 BeeGFS 服務：

```
pcs node standby <HOSTNAME>
```

3. 執行下列步驟，確認節點的服務已耗盡：

```
pcs status
```

驗證沒有任何服務報告為待命節點上的服務 `Started`。

視叢集大小而定， BeeGFS 服務可能需要幾秒鐘或幾分鐘才能移至姊妹節點。如果 BeeGFS 服務無法在姊妹節點上啟動"疑難排解指南"，請參閱。

4. 使用更新介面卡韌體 `mlxfwmanager`。

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

記下 `PCI Device Name`接收韌體更新的每個介面卡的。

5. 使用公用程式重設每個介面卡 `mlxfwreset`以套用新韌體。

某些韌體更新可能需要重新開機才能套用更新。請參閱"NVIDIA 的 mlxfwreset 限制"以取得指引。如果需要重新開機，請執行重新開機，而非重設介面卡。

a. 停止 opensm 服務：

```
systemctl stop opensm
```

b. 針對先前註明的每個項目執行下列命令 `PCI Device Name`。

```
mlxfwreset -d <pci_device_name> reset -y
```

c. 啟動 opensm 服務：

```
systemctl start opensm
```

d. 重新啟動 `eseries_nvme_ib.service` 。

```
systemctl restart eseries_nvme_ib.service
```

e. 驗證 E 系列儲存陣列的磁碟區是否存在。

```
multipath -ll
```

1. 執行 `ibstat`並驗證所有介面卡是否以所需的韌體版本執行：

```
ibstat
```

2. 在節點上啟動 Pacemaker 叢集服務：

```
pcs cluster start <HOSTNAME>
```

3. 將節點從待機狀態移出：

```
pcs node unstandby <HOSTNAME>
```

4. 將所有 BeeGFS 服務重新部署回其偏好的節點：

```
pcs resource relocate run
```

對叢集中的每個檔案節點重複這些步驟，直到所有介面卡都已更新為止。

雙節點叢集更新方法

建議只有兩個節點的 HA 叢集採用此方法。這種方法類似於滾動更新，但包含其他步驟，可在某個節點的叢集服務停止時，避免服務停機。

1. 確認叢集處於最佳狀態，且每個 BeeGFS 服務都在其偏好的節點上執行。如 "檢查叢集的狀態" 需詳細資訊、請參閱。
2. 選擇要更新的檔案節點，並將節點置於待命模式，以從該節點移除（或移動）所有 BeeGFS 服務：

```
pcs node standby <HOSTNAME>
```

3. 執行以下步驟，確認節點的資源已耗盡：

```
pcs status
```

驗證沒有任何服務報告為待命節點上的服務 `Started`。

視叢集大小而定， BeeGFS 服務可能需要幾秒鐘或幾分鐘的時間，才能在姊妹節點上報告為 `Started`。如果 BeeGFS 服務無法啟動，請"疑難排解指南"參閱。

4. 將叢集置於維護模式。

```
pcs property set maintenance-mode=true
```

5. 使用更新介面卡韌體 `mlxfwmanager`。

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

記下 `PCI Device Name`接收韌體更新的每個介面卡的。

6. 使用公用程式重設每個介面卡 `mlxfwreset`以套用新韌體。

某些韌體更新可能需要重新開機才能套用更新。請參閱"NVIDIA 的 mlxfwreset 限制"以取得指引。如果需要重新開機，請執行重新開機，而非重設介面卡。

a. 停止 opensm 服務：

```
systemctl stop opensm
```

b. 針對先前註明的每個項目執行下列命令 `PCI Device Name`。

```
mlxfwreset -d <pci_device_name> reset -y
```

c. 啟動 opensm 服務：

```
systemctl start opensm
```

7. 執行 `ibstat`並驗證所有介面卡是否以所需的韌體版本執行：

```
ibstat
```

8. 在節點上啟動 Pacemaker 叢集服務：

```
pcs cluster start <HOSTNAME>
```

9. 將節點從待機狀態移出：

```
pcs node unstandby <HOSTNAME>
```

10. 將叢集移出維護模式。

```
pcs property set maintenance-mode=false
```

11. 將所有 BeeGFS 服務重新部署回其偏好的節點：

```
pcs resource relocate run
```

對叢集中的每個檔案節點重複這些步驟，直到所有介面卡都已更新為止。

升級 **E-Series** 儲存陣列

請依照下列步驟升級 HA 叢集的 E 系列儲存陣列元件。

總覽

使用最新的韌體、讓 HA 叢集的 NetApp E-Series 儲存陣列保持在最新狀態、以確保最佳效能和更高的安全性。儲存陣列的韌體更新是透過 SANtricity OS ， NVSRAM 和磁碟機韌體檔案來套用。

> ⓘ 雖然儲存陣列可以在 HA 叢集上線時進行升級、但建議您將叢集置於維護模式、以便進行所有升級。

區塊節點升級步驟

下列步驟概述如何使用 `Netapp_Eseries.Santricity` Ansible 集合更新儲存陣列的韌體。在繼續之前、請檢閱"升級考量"以更新 E-Series 系統。

> ⓘ 只能從 11.70.5P1 升級至 SANtricity OS 11.80 或更新版本。在套用進一步升級之前、必須先將儲存陣列升級至 11.70.5P1 。

1. 驗證您的 Ansible 控制節點是否使用最新的 SANtricity Ansible Collection 。
   - 可存取的集合升級 "Ansible Galaxy"，執行下列命令：

```
ansible-galaxy collection install netapp_eseries.santricity --upgrade
```

- 若要進行離線升級"Ansible Galaxy"、請從下載集合 tarball 、將其傳輸至您的控制節點、然後執行：

```
ansible-galaxy collection install netapp_eseries-santricity-
<VERSION>.tar.gz --upgrade
```

請參閱 "安裝集合" 以取得更多資訊。

2. 取得儲存陣列和磁碟機的最新韌體。
   a. 下載韌體檔案。
      - * SANtricity OS 和 NVSRAS:* 瀏覽至"NetApp 支援網站"、並下載適用於您儲存陣列機型的最新版 SANtricity OS 和 NVSRAS* 。
      - * 磁碟機韌體： * 瀏覽"E-Series 磁碟韌體站台"並下載每個儲存陣列磁碟機機型的最新韌體。
   b. 將 SANtricity OS 、 NVSRAM 和磁碟機韌體檔案儲存在 Ansible 控制節點的 `<inventory_directory>/packages`目錄中。
3. 如有必要、請更新叢集的 Ansible 庫存檔案、以納入所有需要更新的儲存陣列（區塊節點）。如需指引、請參閱"Ansible Inventory Overview"一節。
4. 確保叢集處於最佳狀態，且其慣用節點上的每個 BeeGFS 服務都處於最佳狀態。如 "檢查叢集的狀態" 需詳細資訊、請參閱。
5. 按照中的說明將叢集置於維護模式"將叢集置於維護模式"。
6. 建立名為的新 Ansible 教 `update_block_node_playbook.yml`戰手冊。請將下列內容填入教戰手冊、將 SANtricity OS 、 NVSRAM 和磁碟機韌體版本取代為您想要的升級路徑：

```
- hosts: eseries_storage_systems
  gather_facts: false
  any_errors_fatal: true
  collections:
    - netapp_eseries.santricity
  vars:
    eseries_firmware_firmware: "packages/<SantricityOS>.dlp"
    eseries_firmware_nvsram: "packages/<NVSRAM>.dlp"
    eseries_drive_firmware_firmware_list:
      - "packages/<drive_firmware>.dlp"
    eseries_drive_firmware_upgrade_drives_online: true

  tasks:
    - name: Configure NetApp E-Series block nodes.
      import_role:
        name: nar_santricity_management
```

7. 若要啟動更新、請從 Ansible 控制節點執行下列命令：

```
ansible-playbook -i inventory.yml update_block_node_playbook.yml
```

8. 完成教戰手冊後、請確認每個儲存陣列都處於最佳狀態。
9. 將叢集移出維護模式、並驗證叢集處於最佳狀態、每項 BeeGFS 服務都位於其偏好的節點上。

服務與維護

容錯移轉和容錯回復服務

在叢集節點之間移動BeeGFS服務。

總覽

BeeGFS服務可在叢集中的節點之間進行容錯移轉、以確保當節點發生故障或需要執行計畫性維護時、用戶端能夠繼續存取檔案系統。本節說明系統管理員在從故障中恢復後、或在節點之間手動移動服務時、如何修復叢集。

步驟

容錯移轉與容錯回復

容錯移轉（計畫性）

一般而言、當您需要將單一檔案節點離線以進行維護時、您會想要從該節點移動（或耗盡）所有BeeGFS服務。您可以先將節點置於待命狀態、以達成此目標：

```
pcs node standby <HOSTNAME>
```

使用驗證之後 `pcs status` 所有資源都已在替代檔案節點上重新啟動、您可以視需要關機或對節點進行其他變更。

容錯回復（在計畫性容錯移轉之後）

當您準備好將BeeGFS服務還原至首選節點時、請先執行 `pcs status` 並在「Node List（節點清單）」中驗證狀態是否為「standby（待命）」。如果節點重新開機、則會顯示為離線、直到叢集服務上線為止：

```
pcs cluster start <HOSTNAME>
```

節點上線後、請使用以下功能將其從待命模式中移出：

```
pcs node unstandby <HOSTNAME>
```

最後、將所有BeeGFS服務重新部署回其偏好的節點：

```
pcs resource relocate run
```

容錯回復（非計畫性容錯移轉之後）

如果某個節點發生硬體或其他故障、HA叢集應自動回應並將其服務移至正常節點、讓系統管理員有時間採取修正行動。在繼續之前、"疑難排解"請先參閱一節、以判斷容錯移轉的原因、並解決任何未解決的問題。節點重新開機且正常運作後、您就可以繼續進行容錯回復。

當節點在非計畫性（或計畫性）重新開機之後開機時、叢集服務不會設定為自動啟動、因此您必須先使用以下項目使節點上線：

```
pcs cluster start <HOSTNAME>
```

接下來清除任何資源故障並重設節點的隔離記錄：

```
pcs resource cleanup node=<HOSTNAME>
pcs stonith history cleanup <HOSTNAME>
```

請在中驗證 `pcs status` 節點處於線上且健全狀態。根據預設、BeeGFS服務不會自動容錯回復、以避免意外將資源移回不正常的節點。當您準備好時、將叢集中的所有資源、以下列方式傳回其偏好的節點：

```
pcs resource relocate run
```

將個別**BeeGFS**服務移至替代檔案節點

將**BeeGFS**服務永久移至新的檔案節點

如果您想要永久變更個別BeeGFS服務的偏好檔案節點、請調整「Ansible」（可執行）資源清冊、使偏好的節點列在第一位、然後重新執行「Ansible」（可執行）資源清冊。

例如、在此範例檔案中 `inventory.yml` 、 beegfs_01 是執行 BeeGFS 管理服務的慣用檔案節點：

```
        mgmt:
          hosts:
            beegfs_01:
            beegfs_02:
```

反轉訂單會使beegfs_02上的管理服務更受歡迎：

```
        mgmt:
          hosts:
            beegfs_02:
            beegfs_01:
```

暫時將**BeeGFS**服務移至替代檔案節點

一般而言、如果某個節點正在進行維護、您會想要使用[Failover and failover countures]（#Failover與容錯回復步驟）（#容 錯移轉與容錯回復）將所有服務移出該節點。

如果由於某些原因、您確實需要將個別服務移至不同的檔案節點執行：

```
pcs resource move <SERVICE>-monitor <HOSTNAME>
```

> 請勿指定個別資源或資源群組。請務必指定您要重新部署BeeGFS服務的監視器名稱。例如，要將 BeeGFS 管理服務移至 beegfs_02 ，請執行 `pcs resource move mgmt-monitor beegfs_02` ：。您可以重複此程序、將一或多個服務移出偏好的節點。確認 `pcs status` 新節點上的服務已重新定位 / 啟動。

若要將BeeGFS服務移回其慣用節點、請先清除暫用資源限制（視多項服務需要重複此步驟）：

```
pcs resource clear <SERVICE>-monitor
```

當您準備好將服務實際移回偏好的節點時、請執行：

```
pcs resource relocate run
```

請注意、此命令會重新部署任何不再具有暫用資源限制的服務、而這些服務並未位於偏好的節點上。

將叢集置於維護模式

避免HA叢集意外回應環境中的預期變更。

總覽

將叢集置於維護模式會停用所有資源監控、並防止心臟起搏器移動或以其他方式管理叢集中的資源。所有資源都會繼續在其原始節點上執行、無論是否有暫時性的故障情況、都無法存取這些資源。建議/實用的案例包括：

- 可能會暫時中斷檔案節點與BeeGFS服務之間的連線的網路維護。
- 區塊節點升級。
- 檔案節點作業系統、核心或其他套件更新。

一般而言、將叢集手動置於維護模式的唯一理由、是避免它對環境中的外部變更做出回應。如果叢集中的個別節

點需要實體修復、請勿使用維護模式、只要依照上述程序將該節點置於待命狀態即可。請注意、重新執行Ansible會自動將叢集置於維護模式、以利進行大部分的軟體維護、包括升級和組態變更。

步驟

若要檢查叢集是否處於維護模式、請執行：

```
pcs property config
```

`maintenance-mode`如果叢集正常運作，則不會顯示內容。如果叢集目前處於維護模式，則內容會報告為`true`。若要啟用維護模式執行：

```
pcs property set maintenance-mode=true
```

您可以執行PCS狀態並確保所有資源都顯示為「（Unmanaged）」、以進行驗證。若要使叢集離開維護模式執行：

```
pcs property set maintenance-mode=false
```

停止並啟動叢集

正常停止及啟動HA叢集。

總覽

本節說明如何正常關機並重新啟動BeeGFS叢集。可能需要的範例案例包括電力維護或資料中心或機架之間的移轉。

步驟

如果您基於任何原因需要停止整個BeeGFS叢集並關閉所有服務執行：

```
pcs cluster stop --all
```

您也可以在個別節點上停止叢集（這會自動將服務容錯移轉至另一個節點）、不過建議您先將節點置於待命狀態（請參閱"容錯移轉"一節）：

```
pcs cluster stop <HOSTNAME>
```

若要在所有節點上啟動叢集服務和資源、請執行下列步驟：

```
pcs cluster start --all
```

或在特定節點上啟動服務：

```
pcs cluster start <HOSTNAME>
```

此時請執行 `pcs status` 並驗證叢集和BeeGFS服務是否在所有節點上啟動、以及服務是否在您預期的節點上執行。

視叢集大小而定，整個叢集可能需要幾秒鐘或幾分鐘的時間才能停止，或顯示為「已啟動 `pcs status`」。如果當機超過五分鐘，則 `pcs cluster <COMMAND>`在執行「 Ctrl+C 」取消命令之前，請先登入叢集的每個節點，然後使用 `pcs status`查看叢集服務（電暈器同步 / 節律器）是否仍在該節點上執行。從叢集仍在作用中的任何節點、您都可以檢查封鎖叢集的資源為何。手動解決此問題、命令應已完成或可重新執行、以停止任何剩餘的服務。

取代檔案節點

如果原始伺服器故障、請更換檔案節點。

總覽

這是更換叢集中檔案節點所需步驟的總覽。這些步驟假設檔案節點因為硬體問題而失敗、並以新的相同檔案節點取代。

步驟：

1. 實體更換檔案節點、並將所有纜線恢復至區塊節點和儲存網路。
2. 在檔案節點上重新安裝作業系統、包括新增Red Hat訂閱。
3. 在檔案節點上設定管理和BMC網路功能。
4. 如果主機名稱、IP、PCIe對邏輯介面對應、或是新檔案節點的其他任何變更、請更新Ansible詳細目錄。如果節點被相同的伺服器硬體所取代、而且您使用的是原始網路組態、則通常不需要這麼做。
   a. 例如、如果主機名稱已變更、請建立（或重新命名）節點的庫存檔案 (`host_vars/<NEW_NODE>.yml`) 、然後在Ansible庫存檔案中 (`inventory.yml`) 、將舊節點的名稱改為新節點名稱：

```
all:
    ...
    children:
    ha_cluster:
        children:
        mgmt:
            hosts:
            node_h1_new:    # Replaced "node_h1" with "node_h1_new"
            node_h2:
```

5. 從叢集中的其他節點之一移除舊節點： `pcs cluster node remove <HOSTNAME>`。

   執行此步驟之前、請勿繼續。

6. 在Ansible控制節點上：

   a. 移除舊的SSH金鑰：

   ```
   `ssh-keygen -R <HOSTNAME_OR_IP>`
   ```

   b. 將無密碼SSH設定為取代節點：

   ```
   ssh-copy-id <USER>@<HOSTNAME_OR_IP>
   ```

7. 重新執行Ansible playbook以設定節點並將其新增至叢集：

   ```
   ansible-playbook -i <inventory>.yml <playbook>.yml
   ```

8. 此時、請執行 `pcs status` 並確認已列出更換的節點、並正在執行服務。

擴充或縮小叢集

新增或移除叢集的建置區塊。

總覽

本節說明各種考量事項和選項、以調整BeeGFS HA叢集的大小。一般而言、叢集大小是透過新增或移除建置區塊來調整、而建置區塊通常是以HA配對形式設定兩個檔案節點。如果需要、也可以新增或移除個別的檔案節點（或其他類型的叢集節點）。

將建置區塊新增至叢集

考量

透過新增額外的建置區塊來擴充叢集、是一項簡單的程序。開始之前、請記住每個HA叢集中叢集節點的最小和最大數量限制、並決定是否要將節點新增至現有HA叢集、或是建立新的HA叢集。通常每個建置區塊都由兩個檔案節點組成、但每個叢集的節點數最少為三個節點（以建立仲裁）、建議（測試）最多為十個節點。在進階案例中、您可以新增單一「tiebreaker」節點、而在部署雙節點叢集時、該節點不會執行任何BeeGFS服務。如果您正在考慮進行這類部署、請聯絡NetApp支援部門。

在決定如何擴充叢集時、請謹記這些限制和任何預期的未來叢集成長。例如、如果您有六個節點叢集、需要再新增四個節點、建議您只是啟動一個新的HA叢集。

請記住、單一BeeGFS檔案系統可由多個獨立的HA叢集組成。如此一來、檔案系統就能繼續擴充、遠遠超過基礎HA叢集元件的建議/硬限制。

步驟

將建置區塊新增至叢集時、您需要 `host_vars`為每個新的檔案節點和區塊節點（ E-Series 陣列）建立檔案。這些主機的名稱需要新增至庫存、以及要建立的新資源。 `group_vars`需要為每個新資源建立對應的檔案。如"使用自訂架構"需詳細資訊、請參閱一節。

建立正確的檔案之後、只需要使用以下命令重新執行自動化作業：

```
ansible-playbook -i <inventory>.yml <playbook>.yml
```

從叢集移除建置區塊

當您需要淘汰建置區塊時、請謹記下列幾點考量事項：

- 此建置區塊中執行哪些BeeGFS服務？
- 是否只有檔案節點即將淘汰、而且區塊節點應附加至新的檔案節點？
- 如果整個建置區塊都已淘汰、資料是否應該移至新的建置區塊、分散到叢集中的現有節點、或移至新的BeeGFS檔案系統或其他儲存系統？
- 這種情況是否會在停電期間發生、還是應該在不中斷營運的情況下進行？
- 建置區塊是否正在使用中、或主要包含不再使用的資料？

由於可能的起點和所需的終端狀態各不相同、請聯絡NetApp支援部門、以便我們根據您的環境和需求、找出並協助實作最佳策略。

疑難排解

BeeGFS HA叢集疑難排解。

總覽

本節將說明如何調查及疑難排解在操作BeeGFS HA叢集時可能發生的各種故障和其他情況。

疑難排解指南

正在調查意外的容錯移轉

當節點被意外隔離、其服務移至另一個節點時、第一步應該是查看叢集是否在底部顯示任何資源故障 `pcs status`。通常、如果成功完成隔離並在另一個節點上重新啟動資源、則不會出現任何情況。

一般而言、下一步是使用來搜尋整個系統記錄 `journalctl` 在其餘任一檔案節點上（所有節點上的心臟起搏器記錄都會同步）。如果您知道故障發生的時間、可以在故障發生前立即開始搜尋（建議至少提前10分鐘）：

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>"
```

下列各節顯示您可以在記錄中加入的一般文字、以進一步縮小調查範圍。

步驟**1**：檢查**BeeGFS**監視器是否偵測到故障：

如果容錯移轉是由BeeGFS監控器觸發、您應該會看到錯誤（如果沒有、請繼續下一步）。

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i unexpected
[...]
Jul 01 15:51:03 beegfs_01 pacemaker-schedulerd[9246]:  warning: Unexpected
result (error: BeeGFS service is not active!) was recorded for monitor of
meta_08-monitor on beegfs_02 at Jul  1 15:51:03 2022
```

在此例中、BeeGFS服務meta_08因為某些原因而停止。若要繼續疑難排解、我們應該開機 beegfs_02 、並檢閱服務的記錄、網址為： `/var/log/beegfs-meta-meta_08_tgt_0801.log`。例如、BeeGFS服務可能因為內部問題或節點問題而發生應用程式錯誤。

不同於來自心臟起搏器的記錄、BeeGFS服務的記錄不會分散到叢集中的所有節點。若要調查這些故障類型、必須從發生故障的原始節點取得記錄。

監視器可能回報的問題包括：

- 無法存取目標！
  - 說明：表示無法存取區塊磁碟區。
  - 疑難排解：
    - 如果服務也無法在替代檔案節點上啟動、請確認區塊節點正常運作。
    - 請檢查是否有任何實體問題、以免從此檔案節點存取區塊節點、例如發生故障的InfiniBand介面卡或纜線。
- 無法連線到網路！
  - 說明：用戶端用來連線至此BeeGFS服務的介面卡均未連線。
  - 疑難排解：
    - 如果有多個/所有檔案節點受到影響、請檢查網路上是否有故障用於連接BeeGFS用戶端和檔案系統。
    - 請檢查是否有任何實體問題、例如會使此檔案節點無法存取用戶端、例如發生故障的InfiniBand介面卡或纜線。
- BeeGFS服務未啟用！
  - 說明：BeeGFS服務意外停止。
  - 疑難排解：
    - 在報告錯誤的檔案節點上、檢查受影響BeeGFS服務的記錄、以查看是否報告當機。如果發生這種情況、請利用NetApp支援開啟案例、以便調查當機事件。
    - 如果BeeGFS記錄中沒有報告錯誤、請檢查日誌記錄、查看系統是否記錄服務停止的原因。在某些情況下、BeeGFS服務可能沒有機會在程序終止之前記錄任何訊息（例如有人執行） `kill -9 <PID>` ）。

步驟**2**：檢查節點是否意外離開叢集

如果節點發生災難性硬體故障（例如主機板當機）、或發生核心異常或類似軟體問題、BeeGFS監視器將不會報告錯誤。請改為尋找主機名稱、您應該會看到來自心臟起搏器的訊息、指出節點意外遺失：

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i <HOSTNAME>
[...]
Jul 01 16:18:01 beegfs_01 pacemaker-attrd[9245]:  notice: Node beegfs_02
state is now lost
Jul 01 16:18:01 beegfs_01 pacemaker-controld[9247]:  warning:
Stonith/shutdown of node beegfs_02 was not expected
```

步驟**3**：驗證起搏器是否能夠隔離節點

在所有情況下、您都應該看到心臟起搏器試圖將節點隔離、以驗證它實際上是否離線（確切的訊息可能會因隔離原因而異）：

```
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Cluster
node beegfs_02 will be fenced: peer is no longer part of the cluster
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Node
beegfs_02 is unclean
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Scheduling
Node beegfs_02 for STONITH
```

如果隔離動作成功完成、您會看到如下訊息：

```
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]:  notice: Operation 'off'
[2214070] (call 27 from pacemaker-controld.9247) for host 'beegfs_02' with
device 'fence_redfish_2' returned: 0 (OK)
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]:  notice: Operation 'off'
targeting beegfs_02 on beegfs_01 for pacemaker-
controld.9247@beegfs_01.786df3a1: OK
Jul 01 16:18:14 beegfs_01 pacemaker-controld[9247]:  notice: Peer
beegfs_02 was terminated (off) by beegfs_01 on behalf of pacemaker-
controld.9247: OK
```

如果隔離動作因為某種原因而失敗、BeeGFS服務將無法在另一個節點上重新啟動、以避免資料毀損的風險。如果隔離設備（PDU或BMC）無法存取或設定錯誤、則這是需要個別調查的問題。

處理失敗的資源動作（可在**PCS**狀態的底部找到）

如果執行BeeGFS服務所需的資源失敗、BeeGFS監視器會觸發容錯移轉。如果發生這種情況，則可能在底部沒有列出“故障資源操作” `pcs status`，您應該參考有關如何操作的步驟"非計畫性容錯移轉之後的容錯回復"。

否則、通常只會有兩種情況出現「失敗的資源動作」。

案例**1**：使用隔離代理程式偵測到暫時性或永久性問題、然後重新啟動或移至其他節點。

有些屏障代理程式比其他代理程式更可靠、而且每個代理程式都會實作自己的監控方法、以確保屏障設備已就緒。尤其是RedfISH屏障代理程式已被視為回報失敗的資源動作、例如下列動作、即使它仍會顯示為「已啟動」：

```
  * fence_redfish_2_monitor_60000 on beegfs_01 'not running' (7):
call=2248, status='complete', exitreason='', last-rc-change='2022-07-26
08:12:59 -05:00', queued=0ms, exec=0ms
```

回報特定節點上失敗資源動作的隔離代理程式、預期不會觸發該節點上執行BeeGFS服務的容錯移轉。只需在相同或不同的節點上自動重新啟動即可。

解決步驟：

1. 如果隔離代理程式持續拒絕在所有或一部分節點上執行、請檢查這些節點是否能夠連線至隔離代理程式、並確認隔離代理程式已在「Ansible」（可隔離）資源清冊中正確設定。
   a. 例如、如果RedfISH（BMC）屏障代理程式與負責隔離的節點相同、而且OS管理和BMC IP位於同一個實體介面上、則某些網路交換器組態將不允許兩個介面之間進行通訊（以防止網路迴圈）。根據預設、HA叢集會嘗試避免在其負責隔離的節點上放置隔離代理程式、但在某些情況/組態中可能會發生這種情況。
2. 一旦所有問題都解決（或問題似乎是暫時性的）、請執行 `pcs resource cleanup` 以重設失敗的資源動作。

案例**2**：**BeeGFS**監視器偵測到問題並觸發容錯移轉、但由於某些原因、資源無法在次要節點上啟動。

如果啟用了隔離功能、且資源未被封鎖、無法在原始節點上停止（請參閱「待命（故障時）」的疑難排解一節）、最可能的原因包括在次要節點上啟動資源時發生問題、因為：

- 次要節點已離線。
- 實體或邏輯組態問題使次要實體無法存取做為BeeGFS目標的區塊磁碟區。

解決步驟：

1. 針對失敗資源動作中的每個項目：
   a. 確認失敗的資源動作是啟動作業。
   b. 根據指示的資源和故障資源動作中指定的節點：
      i. 尋找並修正任何會使節點無法啟動指定資源的外部問題。例如、如果BeeGFS IP位址（浮動IP）無法啟動、請確認至少有一個必要的介面已連線/連線、並連接至正確的網路交換器。如果BeeGFS目標（區塊裝置/ E系列磁碟區）故障、請驗證與後端區塊節點的實體連線是否如預期連接、並驗證區塊節點是否正常。
   c. 如果沒有明顯的外部問題、而且您想找出造成此事件的根本原因、建議您先向NetApp支援部門提出案例進行調查、然後再繼續進行、因為下列步驟可能會使根本原因分析（RCA）變得具有挑戰性/不可能。
2. 解決任何外部問題之後：

a. 從Ansible inventory.yml檔案中註解任何無法運作的節點、然後重新執行完整的可執行教戰手冊、以確保所有邏輯組態都已在次要節點上正確設定。

   i. 附註：當節點正常運作且您已準備好容錯回復時、請別忘了取消註釋這些節點、然後重新執行教戰手冊。

b. 或者、您也可以嘗試手動恢復叢集：

   i. 使用下列方法將任何離線節點重新連線： `pcs cluster start <HOSTNAME>`

   ii. 使用下列方法清除所有失敗的資源動作： `pcs resource cleanup`

   iii. 執行PCS狀態、並驗證所有服務是否如預期啟動。

   iv. 如有需要、請執行 `pcs resource relocate run` 可將資源移回其首選節點（如果可用）。

常見問題

BeeGFS服務不會在要求時進行容錯移轉或容錯回復

可能的問題： The `pcs resource relocate` 執行命令已執行、但從未成功完成。

*如何檢查：*執行 `pcs constraint --full` 並使用ID檢查任何位置限制 `pcs-relocate-<RESOURCE>`。

*如何解決：*執行 `pcs resource relocate clear` 然後重新執行 `pcs constraint --full` 以驗證是否移除額外的限制。

當隔離功能停用時、**PC**狀態中的一個節點會顯示「待命（故障時）」

*可能的問題：*起搏器無法成功確認故障節點上的所有資源均已停止。

如何解決：

1. 執行 `pcs status` 並檢查是否有任何未「啟動」的資源、或是在輸出底部顯示錯誤、並解決任何問題。
2. 可使節點恢復聯機運行 `pcs resource cleanup --node=<HOSTNAME>`。

在發生非預期的容錯移轉之後、啟用隔離功能時、資源會在**PCS**狀態中顯示「已啟動（故障時）」

*可能的問題：*發生觸發容錯移轉的問題、但心臟起搏器無法驗證節點是否已被隔離。這可能是因為屏障設定錯誤、或屏障代理程式發生問題（例如：PDU已從網路中斷連線）。

如何解決：

1. 驗證節點是否確實關機。

   如果您指定的節點實際上並未關閉、而是執行叢集服務或資源、則會發生資料毀損/叢集故障。

2. 手動確認隔離： `pcs stonith confirm <NODE>`

此時、服務應完成容錯移轉、並在另一個正常節點上重新啟動。

常見疑難排解工作

重新啟動個別**BeeGFS**服務

通常、如果需要重新啟動BeeGFS服務（例如為了協助變更組態）、則應更新「Ansible」（可存取）清單並重新執行播放手冊。在某些情況下、可能需要重新啟動個別服務、以加快疑難排解的速度、例如變更記錄層級、而不需要等待整個方針執行。

除非在Ansible庫存中也新增任何手動變更、否則下次執行Ansible教戰手冊時將會還原這些變更。

選項**1**：系統**d**控制的重新啟動

如果BeeGFS服務可能無法以新組態正確重新啟動、請先將叢集置於維護模式、以防止BeeGFS監視器偵測到服務停止、並觸發不想要的容錯移轉：

```
pcs property set maintenance-mode=true
```

如有需要、請在進行任何服務組態變更 `/mnt/<SERVICE_ID>/_config/beegfs-.conf` （範例：`/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf`）然後使用systemd重新啟動：

```
systemctl restart beegfs-*@<SERVICE_ID>.service
```

範例： `systemctl restart beegfs-meta@meta_01_tgt_0101.service`

選項**2**：心律調整器控制的重新啟動

如果您不擔心新的組態可能會導致服務意外停止（例如、只是變更記錄層級）、或是處於維護期間、不擔心停機、您只需重新啟動BeeGFS監控器、即可取得您要重新啟動的服務：

```
pcs resource restart <SERVICE>-monitor
```

例如、若要重新啟動BeeGFS管理服務： `pcs resource restart mgmt-monitor`

法律聲明

法律聲明提供版權聲明、商標、專利等存取權限。

版權

"https://www.netapp.com/company/legal/copyright/"

商標

NetApp 、 NetApp 標誌及 NetApp 商標頁面上列出的標章均為 NetApp 、 Inc. 的商標。其他公司與產品名稱可能為其各自所有者的商標。

"https://www.netapp.com/company/legal/trademarks/"

專利

如需最新的 NetApp 擁有專利清單、請參閱：

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

隱私權政策

"https://www.netapp.com/company/legal/privacy-policy/"

開放原始碼

通知檔案提供有關 NetApp 軟體所使用之協力廠商版權與授權的資訊。

"E系列/EF系列SANtricity 的注意事項"