使用通過驗證的架構

BeeGFS on NetApp with E-Series Storage

NetApp
June 18, 2024

目錄

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。

此設計充分利用最新企業伺服器與儲存硬體及網路速度所提供的效能密度、 需要具備雙AMD EPYC 7003「Milan」處理器的檔案節點、並支援PCIe 4.0、並使用200GB（HDRs）InfiniBand直接連線至區塊節點、以使用NVMe/IB傳輸協定提供端點對端NVMe和NVMeoF。

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。

使用案例

下列使用案例適用於NetApp上的BeeGFS解決方案：

- 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"。

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"。

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"。

設計世代

BeeGFS on NetApp解決方案目前正處於第二代設計階段。

第一代和第二代均採用基礎架構、整合BeeGFS檔案系統和NVMe EF600儲存系統。然而、第二代產品是以第一代產品為基礎、提供下列額外效益：

- 效能與容量加倍、同時只增加2U機架空間
- 高可用度（HA）是以共享磁碟、雙層硬體設計為基礎
- NVIDIA DGX A100 SupermPOD與NVIDIA BasePOD架構的外部資格

第二代設計

NetApp的第二代BeeGFS已經過最佳化、可滿足嚴苛工作負載的效能需求、包括高效能運算（HPC）和HPC型機器學習（ML）、深度學習（DL）和類似的人工智慧（AI）技術。透過採用共享磁碟高可用度（HA）架構、NetApp上的BeeGFS解決方案也能滿足企業和其他組織的資料耐用度和可用度需求、這些企業和組織在尋找能夠隨工作負載和使用案例而擴充的儲存設備時、無法承受停機或資料遺失的風險。此解決方案不僅已通過NetApp驗證、也已通過外部認證、成為NVIDIA DGX超級POD和DGX基礎POD的儲存選項。

第一代設計

NetApp的第一代BeeGFS是專為使用NetApp EF600 NVMe儲存系統、BeeGFS平行檔案系統、NVIDIA DGX™A100系統及NVIDIA®Mellanox®Quantum™QM8700 200Gbps IB交換器的機器學習（ML）和人工智慧（AI）工作負載所設計。此設計也針對儲存設備和運算叢集互連架構提供200Gbps InfiniBand（IB）、為高效能工作負載提供完全以IB為基礎的架構。

如需第一代產品的詳細資訊、請參閱 "NetApp EF系列AI搭配NVIDIA DGX A100系統和BeeGFS"。

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。

已驗證節點

NetApp上的BeeGFS解決方案包含下列通過驗證的節點：NetApp EF600儲存系統（區塊節點）和Lenovo ThinkSystem SR665 Server（檔案節點）。

區塊節點：**EF600**儲存系統

NetApp EF600 All Flash Array提供一致、近乎即時的資料存取、同時支援任意數量的工作負載。EF600儲存系統為AI和HPC應用程式提供快速且持續的資料饋送功能、可在單一機箱中提供高達200萬個快取讀取IOPS、100微秒以下的回應時間、以及42GBps循序讀取頻寬。

檔案節點：**Lenovo ThinkSystem SR665**伺服器

SR665是雙插槽2U伺服器、搭載PCIe 4.0。當設定為符合此解決方案的需求時、它可提供充分的效能、在組態中執行BeeGFS檔案服務、並與直接附加E系列節點所提供的處理量和IOPs的可用度保持良好平衡。

如需Lenovo SR665的詳細資訊、請參閱 "聯想的網站"。

已驗證硬體設計

解決方案的建置區塊（如下圖所示）使用雙插槽PCIe 4.0伺服器作為BeeGFS檔案層、使用兩個EF600儲存系統作為區塊層。

[]

由於每個建置區塊都包含兩個BeeGFS檔案節點、因此在容錯移轉叢集中建立仲裁所需的建置區塊至少要有兩個。雖然您可以設定雙節點叢集、但此組態具有限制、可能會導致容錯移轉失敗。如果您需要雙節點叢集、可以將第三個裝置整合為斷路器（不過、此站台並未涵蓋該設計）。

每個建置區塊都能透過雙層硬體設計、將檔案層和區塊層的故障網域區隔開來、提供高可用度。每個層級都能獨立容錯移轉、提供更高的恢復能力、並降低串聯故障的風險。搭配NVMeoF使用HDRInfiniBand可在檔案和區塊節點之間提供高處理量和最低延遲、並提供完整備援和足夠的連結超額訂閱、以避免分散式設計成為瓶頸、即使系統部分降級。

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都是透過軟體設定、以執行BeeGFS中繼資料和儲存服務、或僅執行儲存服務。每個建置區塊的不同組態設定檔可用度、可利用相同的基礎硬體平台和建置區塊設計來擴充檔案系統中繼資料或儲存容量和效能。

獨立式Linux HA叢集最多可合併五個建置區塊、確保每個叢集資源管理程式（起搏器）擁有合理數量的資源、並減少維持叢集成員同步所需的訊息傳輸負荷（電暈器同步）。建議每個叢集至少有兩個建置區塊、以允許足夠的成員建立仲裁。其中一個或多個獨立式BeeGFS HA叢集會結合在一起、以建立一個BeeGFS檔案系統（如下圖所示）、可供用戶端作為單一儲存命名空間存取。

[]

雖然每個機架的建置區塊數量最終取決於特定站台的電力和冷卻需求、 此解決方案的設計可在單一42U機架中部署最多五個建置區塊、同時仍可容納兩個用於儲存/資料網路的1U InfiniBand交換器。每個建置區塊都需要八個IB連接埠（每個交換器四個用於備援）、因此五個建置區塊會在40埠的HDRInfiniBand交換器（例如NVIDIA QM8700）上留下一半的連接埠、以供實作fat樹狀結構或類似的非區塊拓撲。這項組態可確保儲存設備或運算/GPU機架的數量能夠擴充、而不會產生網路瓶頸。或者、您也可以根據儲存架構廠商的建議、使用超額訂閱的儲存架構。

下圖顯示80節點的fat樹狀結構拓撲。

[]

藉由使用Ansible做為部署引擎、在NetApp上部署BeeGFS、系統管理員可以使用現代化的基礎架構做為程式碼

實務、來維護整個環境。如此可大幅簡化原本複雜的系統、讓系統管理員能夠在同一個位置定義及調整組態、無論環境的規模有多大、都能確保系統一致地套用組態。BeeGFS系列可從取得 "Ansible Galaxy" 和 "NetApp的E系列GitHub"。

技術需求

若要在NetApp上實作BeeGFS解決方案、請確定您的環境符合技術需求。

硬體需求

下表列出在NetApp解決方案上實作單一第二代建置區塊設計所需的硬體元件。

解決方案的任何特定實作所使用的硬體元件、可能會因客戶需求而異。

數	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下列組態、以達到預期的效能。 處理器： • 2個AMD EPYC 7343 16C 3.2 GHz。 • 設定為兩個NUMA區域。 記憶體： • 256GB。 • 16x 16GB Truc4 3200MHz（2RX8 1.2V）RDIMM A（較少的DIMM容量更小）。 • 以最大化記憶體頻寬。 • PCIe擴充：四個PCE Gen4 x16插槽：* • 每個NUMA區域有兩個插槽。 • 每個插槽都應為Mellanox MCX653106A-HDAT介面卡提供足夠的電力/冷卻能力。 雜項： • 在 RAID 1 中為作業系統設定兩個 1TB 7.2K SATA 磁碟機（或同等或更高）。 • 1 GbE （或更好的）連接埠、用於頻內 OS 管理。 • 1GbE BMC搭配Redfish API、可進行頻外伺服器管理。 • 雙熱交換電源供應器與效能風扇。 • 如果需要連接儲存InfiniBand交換器、則必須支援Mellanox光纖InfiniBand纜線。 聯想**SR665-** • 自訂的NetApp機型包括所需的XClarity控制器韌體版本、以支援雙埠Mellanox ConnectX-6介面卡。如需訂購詳細資料、請聯絡NetApp。
8.	Mellanox ConnectX-6 HCA（適用於檔案節點）。	• MCX653106A-HDAT主機通道配接卡（HDRB 200GB、雙埠QSFP56、PCIe4.0 x16）。
8.	1M HDRInfiniBand纜線（適用於檔案/區塊節點直接連接）。	• MCP1650-H001E30（1公尺Mellanox被動銅線、IB HDR、最高200Gbps、QSFP56、30AWE）。 如果需要、可調整長度以考慮檔案與區塊節點之間的距離。

數	硬體元件	需求
8.	HDRInfiniBand纜線（適用於檔案節點/儲存交換器連線）	需要長度適當的InfiniBand HDRdr纜線（QSFP56收發器）、才能將檔案節點連接至儲存葉交換器。可能的選項包括： • MCP1650-H002E26（2公尺Mellanox被動銅線、IB HDRs、最高200GB/s、QSFP56、30AWE）。 • MSFS1S00-H003E（3公尺Mellanox主動式光纖纜線、IB HDRs、最高200GB/s、QSFP56）。
2.	E系列區塊節點	兩個EF600控制器設定如下： • 記憶體：256GB（每個控制器128GB）。 • 介面卡：2埠200GB/HDR（NVMe / IB）。 • 磁碟機：設定以符合所需容量。

軟體需求

為了達到可預測的效能與可靠性、NetApp解決方案上的BeeGFS版本會測試採用實作解決方案所需的特定軟體元件版本。

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	7.2.6
電暈器同步	3.1.5-1.
起搏器	2.1.0-8
OpenSM	OpenSM-5.9.0（從mlnx_ofed 5.4.0-1.3.0） 只有直接連線才需要啟用虛擬化。

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	2.11透過pip安裝時：Ansible -4.7.0與Ansible -core < 2.12、>=2.11.6
Python	3.9
其他Python套件	密碼編譯-35.5.0、netaddr-0.8.0

軟體	版本
BeeGFS Ansible收藏	3.0.00.0

檔案節點需求

軟體	版本
RedHat Enterprise Linux	RedHat 8.4伺服器實體配備高可用度（2插槽）。 檔案節點需要有效的RedHat Enterprise Linux Server訂閱和Red Hat Enterprise Linux高可用度附加元件。
Linux核心	4.18.0-305.25.1.el8_4.x86_64
InfiniBand / RDMA驅動程式	收件匣
ConnectX-6 HCA韌體	韌體：20.31.1014
PXE-3.6.0403.	UEFI：14.24.0013

EF600區塊節點需求

軟體	版本
作業系統SANtricity	11.70.2
NVSRAM	N6000-872834 - D06.dlp
磁碟機韌體	最新版本適用於使用中的磁碟機機型。

其他需求

下表所列設備用於驗證、但可視需要使用適當的替代方案。一般而言、NetApp建議執行最新的軟體版本、以避免非預期的問題。

硬體元件	安裝軟體
• 2個Mellanox MQM8700 200GB InfiniBand交換器	• 韌體3.9.2110
• 1個可控制節點（虛擬化）：* • 處理器：Intel（R）Xeon（R）Gold 6146 CPU @ 3.20GHz • 記憶體：8GB • 本機儲存設備：24GB	• CentOS Linux 8.4.2105 • 核心4.18.0-305.3.1.el8.x86_64 安裝的Ansible和Python版本與上表中的版本相符。

硬體元件	安裝軟體
• 10倍BeeGFS用戶端（CPU節點）：* • 處理器：1個AMD EPYC 7302 16核心CPU（3.0GHz） • 記憶體：128GB • 網路：2個Mellanox MCX653106A-HDAT（每個介面卡連接一個連接埠）。	• Ubuntu 20.04 • 核心：5.4.0-100-generic • InfiniBand驅動程式：Mellanox OFED 5.4.1到1.0.3.0
• 1個BeeGFS用戶端（GPU節點）：* • 處理器：2個採用2.25GHz的AMD EPYC 7742 64核心CPU • 記憶體：1TB • 網路：2個Mellanox MCX653106A-HDAT（每個介面卡連接一個連接埠）。 此系統以NVIDIA HGX A100平台為基礎、內含四個A100 GPU。	• Ubuntu 20.04 • 核心：5.4.0-100-generic • InfiniBand驅動程式：Mellanox OFED 5.4.1到1.0.3.0

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。

[]

對於具有備援交換器的儲存網路、以淺綠色列出的連接埠應連接至一台交換器、以深綠色標示的連接埠則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。理論上、HDRInfiniBand連接埠的最大速度為25Gbps（不會計訊號和其他負荷）。PCIe 4.0 x16插槽的最大單一方向頻寬為32Gbps、因此在實作包含雙埠InfiniBand介面卡的檔案節點時、可能會造成瓶頸、理論上可處理50GBps的頻寬。

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獨立網路「軌道」來提高效能的能力。

  雖然BeeGFS可以使用RDMA進行連線、但BeeGFS使用IPoIB來簡化探索和建立RDMA連線的程序。若要允許BeeGFS用戶端使用多個InfiniBand介面、您可以使用位於不同子網路中的IP位址來設定每個用戶端、然後為每個子網路中一半的BeeGFS伺服器服務設定偏好的介面。

  在下圖中、以淺綠色強調顯示的介面位於一個IP子網路（例如「100127.0.0/16」）、而暗綠色介面則位於另一個子網路（例如「100128.0.0/16」）。

  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@ictad21h01:/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]
```

使用兩個IPoIB子網路是邏輯上的差異。您可以視需要使用單一實體InfiniBand子網路（儲存網路）。

> BeeGFS 7.3.0新增多重軌道支援、可在單一IPoIB子網路中使用多個IB介面。BeeGFS on NetApp解決方案的設計是在BeeGFS 7.3.0正式推出之前開發、因此示範如何使用兩個IP子網路來使用BeeGFS用戶端上的兩個IB介面。多重IP子網路方法的一項優點是不需要在BeeGFS用戶端節點上設定多重主頁（如需詳細資訊、請參閱） "BeeGFS RDMA支援"）。

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。如需詳細資訊、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。

建置區塊連接至兩個Mellanox 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。

[]

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。

[]

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 A100 SupermPOD與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。所有通過NVIDIA DGX A100 SupermPOD認證的儲存設備、都會自動通過NVIDIA BasePOD架構的使用認證。

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。若要瞭解如何設定已驗證之Lenovo SR665檔案節點的系統設定、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」清單時、可以覆寫預設的Linux OS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Mellanox HCA的最大讀取要求大小。

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

您可以使用第二代NetApp BeeGFS建置區塊設計、在NetApp上部署BeeGFS至已驗證的檔案和區塊節點。

Ansible集合與角色

您可以使用Ansible（常用的IT自動化引擎）在NetApp上部署BeeGFS解決方案、以自動化應用程式部署。Ansible會使用一系列的檔案、統稱為庫存、用來建立您要部署的BeeGFS檔案系統模型。

Ansible允許NetApp等公司使用Ansible銀河系統上的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的模組（例如建立E系列Volume）、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、它還能簡化新增建置區塊的作業、以擴充現有的檔案系統。

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建置區塊的數量、可能會因程序而異、視容量和效能需求而定。

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。請密切注意任何應針對特定部署進行修改或可修改之字元以「#」或「//」表示的即時註解。例如：「beegfs_ha_ntp_server_Pools：#這是註解的範例！-「Pool 0.pool.ntp.org ipburst連串最大資源3」-「Pool 1.pool.ntp.org ipburst最大資源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

開始部署之前、請務必瞭解如何使用Ansible、使用第二代BeeGFS建置區塊設計、在NetApp上設定及部署BeeGFS解決方案。

Ansible清單定義檔案和區塊節點的組態、並代表您要部署的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。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集合中提供的各種Ansible模組和角色、尤其是BeeGFS HA 7.2角色（可從取得 "NetApp E系列BeeGFS GitHub"）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」（可執行）庫存結構來定義BeeGFS HA叢集。

任何具備前述經驗的人都應該知道、BeeGFS HA角色會執行自訂方法、以探索哪些變數（或事實）適用於每個主機。這是為了簡化建立可執行的資源庫存、以說明可在多個伺服器上執行的資源。

Ansible清單通常包含「host_vars」和「Group_vars」中的檔案、以及指派主機給特定群組（以及可能指派給其他群組的群組）的「inventory.yml」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:
        ictad22a01:
        ictad22a02:
        ictad22a03:
        ictad22a04:
        ictad22a05:
        ictad22a06:
    # Ansible group representing all file nodes:
    ha_cluster:
      children:
        meta_01:  # Group representing a metadata service with ID 01.
          hosts:
            file_node_01:  # This service is preferred on the first file
node.
            file_node_02:  # And can failover to the second file node.
        meta_02:  # Group representing a metadata service with ID 02.
          hosts:
            file_node_02:  # This service is preferred on the second file
node.
            file_node_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>
  - i4b: <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:
  ictad22a01:
    eseries_storage_pool_configuration:
      - name: beegfs_m1_m2_m5_m6
        raid_level: raid1
        criteria_drive_count: 4
        owning_controller: A
        common_volume_configuration:
          segment_size_kb: 128
        volumes:
          - size: 21.25
```

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。如需詳細資訊、請參閱以下部落格文章： "NetApp利用Ansible加速部署HA for BeeGFS"。

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（位置）[ROW][RAG] HN（機架）」看起來像「ictad22h01」。

- 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、名為「ictad22a01」的區塊節點通常可以為每個控制器設定主機名稱、例如「ictad22a01-a」和「ictad22a01-b」、但在「Ansible」目錄中稱為「ictad22a01」。

- 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（兩個子網路）

為了讓BeeGFS用戶端使用兩個InfiniBand連接埠、每個子網路需要兩個IPoIB子網路、而BeeGFS伺服器服務的一半則在每個子網路上設定偏好的IP、以確保用戶端使用兩個InfiniBand連接埠來最大化備援和檔案系統的可能處理量。

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儲存服務永遠是「xxx.xxx.y.103.zzz」或「xxx.xxx.y.103.zzz」。
- 範圍從「100.xxx.1.1」到「100.xxx.99.255」的位址會保留給用戶端。

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
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。

由於每個建置區塊都包含兩個BeeGFS檔案節點、因此在容錯移轉叢集中建立仲裁所需的建置區塊至少要有兩個。雖然可以設定雙節點叢集、但此組態有一些限制、可能會在某些情況下導致容錯移轉失敗。如果需要雙節點叢集、也可以將第三個裝置整合為斷路器、但此部署程序並未涵蓋這一點。

除非另有說明、否則無論是用於執行BeeGFS中繼資料和儲存服務、或僅用於儲存服務、叢集中每個建置區塊的下列步驟都相同。

步驟

1. 以InfiniBand模式設定每個BeeGFS檔案節點、並在PCIe插槽2、3、5及6中安裝四個PCIe 4.0 ConnectX-6雙埠主機通道介面卡（HCA）。
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（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. 若要纜線連接檔案和區塊節點、請使用1公尺InfiniBand HDR200GB直接連接銅線、使其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2公尺（或適當長度）的InfiniBand HDR200GB直接連接銅線、將每個檔案節點上剩餘的InfiniBand連接埠纜線連接至儲存網路所用的InfiniBand交換器。

如果使用的是冗餘InfiniBand交換器、請將下圖中以淺綠色強調顯示的連接埠連接至不同的交換器。

[]

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。

   若要瞭解如何設定已驗證之Lenovo SR665檔案節點的系統設定、請參閱 "調整系統設定以獲得效能"。

3. 安裝Red Hat 8.4並設定用於管理作業系統的主機名稱和網路連接埠、包括從Ansible控制節點進行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RedHat訂購管理程式來註冊及訂閱系統、以便從正式的Red Hat儲存庫安裝所需的套件、並將更新限制為支援的Red Hat版本：「Subscree-manager版本—set=8.4」。如需相關指示、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-8-for-x86_64
-highavailability-rpms --add=enabled:1
```

6. 將所有ConnectX-6 HCA韌體更新為中建議的版本 "技術需求"。

   您可以下載並執行已套用建議韌體的mlxup工具版本、以完成此更新。您可以從下載此工具 "mlxup -更新與查詢公用程式" （"使用指南"）。

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。

   如需設定連接埠的指示、請前往 "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。如需設定陣列名稱的指示、請前往 "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。

設定可**Ansible**控制節點

若要設定Ansible控制節點、您必須識別可透過網路存取所有檔案和區塊節點管理連接埠的虛擬或實體機器、以便用來設定解決方案。

下列步驟已在CentOS 8.4上進行測試。如需您偏好的Linux套裝作業系統專屬的步驟、請參閱 "Ansible文件"。

1. 安裝Python 3.9並確保安裝正確版本的「pip」。

```
sudo dnf install python3.9 -y
sudo dnf install python39-pip
sudo dnf install sshpass
```

2. 建立符號連結、確保每當叫用「python3」或「python」時、都會使用Python 3.9二進位檔。

```
sudo ln -sf /usr/bin/python3.9 /usr/bin/python3
sudo ln -sf /usr/bin/python3 /usr/bin/python
```

3. 安裝NetApp BeeGFS集合所需的Python套件。

```
python3 -m pip install ansible cryptography netaddr
```

> 若要確保您安裝的是受支援版本的Ansible和所有必要的Python套件、請參閱BeeGFS集合的讀我檔案。支援的版本也會在中註明 "技術需求"。

4. 確認已安裝Ansible和Python的正確版本。

```
ansible --version
ansible [core 2.11.6]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.9/site-
packages/ansible
  ansible collection location =
/root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.9.2 (default, Mar 10 2021, 17:29:56) [GCC 8.4.1
20200928 (Red Hat 8.4.1-1)]
  jinja version = 3.0.2
  libyaml = True
```

5. 將用來描述BeeGFS部署的Ansible庫存儲存在Git或BitBucket等來源控制系統中、然後安裝Git與這些系統互動。

```
sudo dnf install git -y
```

6. 設定無密碼SSH。這是讓Ansible從「Ansible」控制節點存取遠端BeeGFS檔案節點最簡單的方法。
   a. 如果需要、在Ansible控制節點上、使用「sh-keygen'」產生一對公開金鑰
   b. 使用「sh-copy -id <ip_or_hostname>'」設定每個檔案節點的無密碼SSH

      請*不要*設定區塊節點的無密碼SSH。這既不受支援、也不需要。

7. 使用Ansible銀河來安裝中所列BeeGFS集合的版本 "技術需求"。

   此安裝包括其他可相依的相依性、例如NetApp SANtricity 整套軟體和主機收藏。

```
ansible-galaxy collection install netapp_eseries.beegfs:==3.0.1
```

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

- 使用BitBucket或Git等來源控制系統來儲存內含Ansible庫存和教戰手冊檔案的目錄內容。
- 建立一個.gidignore檔案、指定Git應該忽略的檔案。這有助於避免在Git中儲存大型檔案。

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"

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: 100.128.100. <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: root
ansible_become_password: <PASSWORD>
eseries_ipoib_default_hook_templates:
  - 99-multihoming.j2 # This is required when configuring additional
static IPs (for example cluster IPs) when multiple IB ports are in the
same IPoIB subnet.
# 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 --state=active,exited |
grep eseries_nvme_ib.service"
```

> 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"）或是執行教戰手冊時的「Ask–變為pass」選項。如果「Ansip_ssh使用者」已經是「root」、您可以選擇省略「Ansible變成密碼」。

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: 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.
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: <PDU_USERNAME>
     passwd: <PDU_PASSWORD>
     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>
  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
# For details on configuring other fencing agents see
https://access.redhat.com/documentation/en-
us/red_hat_enterprise_linux/8/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。

   如需可調整的可用調校參數完整清單、請參閱BeeGFS HA角色的效能調校預設值一節 "E系列BeeGFS GitHub網站"。此檔案中的叢集所有節點或個別節點的「host_vars」檔案、都可以覆寫預設值。

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Mellanox Open Fabrics Enterprise Distribution（MLNX_OFED）中的Open Subnet Manager（OpenSM）套件。（收件匣「opensm」套件不支援必要的虛擬化功能。） 雖然支援使用Ansible進行部署、但您必須先將所需的套件下載至執行BeeGFS角色所用的Ansible控制節點。

   a. 使用「Curl」或您想要的工具、從Mellanox的網站下載列在技術需求區段中的OpenSM版本套件至「套件/」目錄。例如：

   ```
   curl -o packages/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
   0.1.54103.x86_64.rpm
   https://linux.mellanox.com/public/repo/mlnx_ofed/5.4-
   1.0.3.0/rhel8.4/x86_64/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
   0.1.54103.x86_64.rpm

   curl -o packages/opensm-5.9.0.MLNX20210617.c9f2ade-
   0.1.54103.x86_64.rpm
   https://linux.mellanox.com/public/repo/mlnx_ofed/5.4-
   1.0.3.0/rhel8.4/x86_64/opensm-5.9.0.MLNX20210617.c9f2ade-
   0.1.54103.x86_64.rpm
   ```

   b. 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_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-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm": "/tmp/"
          "packages/opensm-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm": "/tmp/"
    - packages:
        add:
          - /tmp/opensm-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm
          - /tmp/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm
  uninstall:
    - packages:
        remove:
          - opensm
          - opensm-libs
      files:
        remove:
          - /tmp/opensm-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm
          - /tmp/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm
eseries_ib_opensm_options:
  virt_enabled: "2"
```

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 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

# Note: At this time no other x86 servers have been qualified.
Configuration for future qualified file nodes will be added here.
```

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: <PASSWORD>
eseries_validate_certs: false
```

不建議以純文字列出任何密碼。使用Ansible保存庫或使用「-Extra vars」執行Ansible時提供「Eserie_system_password」。

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_11.70.2_6000_61b1131d.dlp"
eseries_firmware_nvsram: "packages/N6000-872834-D06.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、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：

```
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:
        ictad22a01:
        ictad22a02:
        ictad22a03:
        ictad22a04:
        ictad22a05:
        ictad22a06:
    # 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：

```
      # ictad22h01/ictad22h02 HA Pair (mgmt/meta/storage building
block):
        mgmt:
          hosts:
            ictad22h01:
            ictad22h02:
        meta_01:
          hosts:
            ictad22h01:
            ictad22h02:
        stor_01:
          hosts:
            ictad22h01:
            ictad22h02:
        meta_02:
          hosts:
            ictad22h01:
            ictad22h02:
        stor_02:
          hosts:
```

```
        ictad22h01:
        ictad22h02:
    meta_03:
      hosts:
        ictad22h01:
        ictad22h02:
    stor_03:
      hosts:
        ictad22h01:
        ictad22h02:
    meta_04:
      hosts:
        ictad22h01:
        ictad22h02:
    stor_04:
      hosts:
        ictad22h01:
        ictad22h02:
    meta_05:
      hosts:
        ictad22h02:
        ictad22h01:
    stor_05:
      hosts:
        ictad22h02:
        ictad22h01:
    meta_06:
      hosts:
        ictad22h02:
        ictad22h01:
    stor_06:
      hosts:
        ictad22h02:
        ictad22h01:
    meta_07:
      hosts:
        ictad22h02:
        ictad22h01:
    stor_07:
      hosts:
        ictad22h02:
        ictad22h01:
    meta_08:
      hosts:
        ictad22h02:
        ictad22h01:
```

```
        stor_08:
          hosts:
            ictad22h02:
            ictad22h01:
```

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: 100.128.102.0/16
beegfs_service: management
beegfs_targets:
  ictad22a01:
    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、128.102.1/16	0	ictad22a01	beegfs_m1_m2_m5_m6.	答
meta_02.yml	8025	i2b:100128.102.2/16 i1b:100127.101.2/16	0	ictad22a01	beegfs_m1_m2_m5_m6.	b
meta_03.yml	8035	i3b：100127.101.3/16 i4b：100128.102.3/16	1.	ictad22a02	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、128.102.4 / 16 i3b：100、127.101.4 / 16	1.	ictad22a02	Beegfs_m3_m4_m7_M8	b
meta_05.yml	8055	i1b：100127.101.5/16 i2b：100128.102.5/16	0	ictad22a01	beegfs_m1_m2_m5_m6.	答
meta_06.yml	8065	i2b:100128.102.6/16 i1b:100127.101.6/16	0	ictad22a01	beegfs_m1_m2_m5_m6.	b
meta_07.yml	8075	i3b：100、127.101.7 / 16 i4b：100、128.102.7 / 16	1.	ictad22a02	Beegfs_m3_m4_m7_M8	答
meta_08.yml	8085	i4b：100、128.102.8 / 16 i3b：100、127.101.8 / 16	1.	ictad22a02	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、128.104.1/16	0	ictad22a01	beegfs_s1_s2	答
STOR_02.yml	8023	i2b:100128.104.2/16 i1b:100127.103.2/16	0	ictad22a01	beegfs_s1_s2	b
STOR_03.yml	8033	i3b：100、127.103.3 / 16 i4b：100、128.104.3/16	1.	ictad22a02	beegfs_s2_s4	答
STOR_04.yml	8043	i4b：100、128.104.4/16 i3b：100、127.103.4/16	1.	ictad22a02	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、128.104.5/16	0	ictad22a01	Beegfs_S1_S6	答
STOR_06.yml	8063	i2b:100128.104.6/16 i1b:100127.103.6/16	0	ictad22a01	Beegfs_S1_S6	b
STOR_07.yml	8073	i3b：100、127.103.7 / 16 i4b：100、128.104.7/16	1.	ictad22a02	Beegfs_S7_S8	答
STOR_08.yml	8083	i4b：100、128.104.8/16 i3b：100、127.103.8 / 16	1.	ictad22a02	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:
            ictad22h03:
            ictad22h04:
        stor_09:
          hosts:
            ictad22h03:
            ictad22h04:
        meta_10:
          hosts:
            ictad22h03:
            ictad22h04:
        stor_10:
          hosts:
            ictad22h03:
            ictad22h04:
        meta_11:
          hosts:
```

```
        ictad22h03:
        ictad22h04:
    stor_11:
      hosts:
        ictad22h03:
        ictad22h04:
    meta_12:
      hosts:
        ictad22h03:
        ictad22h04:
    stor_12:
      hosts:
        ictad22h03:
        ictad22h04:
    meta_13:
      hosts:
        ictad22h04:
        ictad22h03:
    stor_13:
      hosts:
        ictad22h04:
        ictad22h03:
    meta_14:
      hosts:
        ictad22h04:
        ictad22h03:
    stor_14:
      hosts:
        ictad22h04:
        ictad22h03:
    meta_15:
      hosts:
        ictad22h04:
        ictad22h03:
    stor_15:
      hosts:
        ictad22h04:
        ictad22h03:
    meta_16:
      hosts:
        ictad22h04:
        ictad22h03:
    stor_16:
      hosts:
        ictad22h04:
        ictad22h03:
```

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、128.102.9 / 16	0	ictad22a03	Beegfs_m9_m10_M13_M14	答
meta_10.yml	8025	i2b:100128.102.10/16 i1b:100127.101.10/16	0	ictad22a03	Beegfs_m9_m10_M13_M14	b
meta_11.ml	8035	i3b：100、127.101.11 / 16 i4b：100、128.102.11/16	1.	ictad22a04	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、128.102.12/16 i3b：100、127.101.12 / 16	1.	ictad22a04	Beegfs_M11_M12_M15_M16	b
meta_13.yml	8055	i1b：100、127.101.13 / 16 i2b：100、128.102.1/16	0	ictad22a03	Beegfs_m9_m10_M13_M14	答
meta_14.yml	8065	i2b:100128.102.14/16 i1b:100127.101.14/16	0	ictad22a03	Beegfs_m9_m10_M13_M14	b
meta_15.yml	8075	i3b：100、127.101.15、16 i4b：100、128.102.15/16	1.	ictad22a04	Beegfs_M11_M12_M15_M16	答
meta_16.myl	8085	i4b：100、128.102.16/16 i3b：100、127.101.16/16	1.	ictad22a04	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>
```

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_09.yml	8013	i1b：100、127.103.9/16 i2b：100、128.104.9 / 16	0	ictad22a03	beegfs_s9_s10	答
STOR_10.yml	8023	i2b:100128.104.10/16 i1b:100127.103.10/16	0	ictad22a03	beegfs_s9_s10	b
STOR_11.yml	8033	i3b：100、127.103.11/16 i4b：100、128.104.11/16	1.	ictad22a04	Beegfs_S11_s12.	答
Stor_12.ml	8043	i4b：100、128.104.12/16 i3b：100、127.103.12/16	1.	ictad22a04	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、128.104.13/16	0	ictad22a03	beegfs_s13_s14	答
STOR_14.yml	8063	i2b:100128.104.14/16 i1b:100127.103.14/16	0	ictad22a03	beegfs_s13_s14	b
STOR_15.yml	8073	i3b：100、127.103.15/16 i4b：100、128.104.15/16	1.	ictad22a04	Beegfs_S15_S16	答
STOR_16.yml	8083	i4b：100、128.104.16/16 i3b：100、127.103.16/16	1.	ictad22a04	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：

```
# ictad22h05/ictad22h06 HA Pair (storage only building block):
  stor_17:
    hosts:
      ictad22h05:
      ictad22h06:
  stor_18:
    hosts:
      ictad22h05:
      ictad22h06:
  stor_19:
    hosts:
      ictad22h05:
      ictad22h06:
  stor_20:
    hosts:
      ictad22h05:
      ictad22h06:
  stor_21:
    hosts:
      ictad22h06:
      ictad22h05:
  stor_22:
    hosts:
      ictad22h06:
      ictad22h05:
  stor_23:
    hosts:
      ictad22h06:
      ictad22h05:
  stor_24:
    hosts:
      ictad22h06:
      ictad22h05:
```

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>
```

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_17.yml	8013	i1b：100、127.103.17/16 i2b：100、128.104.17/16	0	ictad22a05	Beegfs_S17_s18	答
STOR_18.yml	8023	i2b:100128.104.18/16 i1b:100127.103.18/16	0	ictad22a05	Beegfs_S17_s18	b
STOR_19.yml	8033	i3b：100127.103.19/16 i4b：100128.104.19/16	1.	ictad22a06.	Beegfs_s19_S20	答
Stor_20.ml	8043	i4b：100、128.104.20/16 i3b：100、127.103.20/16	1.	ictad22a06.	Beegfs_s19_S20	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_21.yml	8053	i1b：100、127.103.21/16 i2b：100、128.104.21/16	0	ictad22a05	Beegfs_S21_S22	答
STOR_22.yml	8063	i2b:100128.104.22/16 i1b:100127.103.22/16	0	ictad22a05	Beegfs_S21_S22	b
STOR_23.yml	8073	i3b：100、127.103.23 / 16 i4b：100、128.104.23 / 16	1.	ictad22a06.	beegfs_S23_s24	答
STOR_24.yml	8083	i4b：100、128.104.24/16 i3b：100、127.103.24/16	1.	ictad22a06.	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.
      import_role:
        name: beegfs_ha_7_2
```

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、建議使用「叉具」參數來覆寫預設的叉具數量（5）、以增加可選主機平行設定的主機數量。（如需詳細資訊、請參閱 "Ansible Performance調校" 和 "控制方針執行"） 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>0
    address: <IP>/<SUBNET_MASK>
```

目前、每個用戶端都必須設定兩個InfiniBand介面、兩個儲存IPoIB子網路各一個。如果針對此處列出的每個BeeGFS服務使用範例子網路和建議範圍、用戶端應該設定一個介面、範圍為「100127.1」。從0到100、127.99.255、另一個在100、128.1。0到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:
        ictad21h01:
        ictad21h02:
        ictad21h03:
        ictad21h04:
        ictad21h05:
        ictad21h06:
        ictad21h07:
        ictad21h08:
        ictad21h09:
        ictad21h10:
      vars:
        # OPTION 1: If you're using the Mellanox 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Mellanox OFED驅動程式時、請確定「beegfs_client_ofed_include_path」指向適用於Linux安裝的正確「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"。

版權資訊

商標資訊