



使用自訂架構 BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

目錄

使用自訂架構	1
總覽與需求	1
簡介	1
部署總覽	1
需求	1
初始設定	2
安裝及纜線硬體	2
設定檔案和區塊節點	6
設定Ansible Control Node	7
定義BeeGFS檔案系統	7
Ansible Inventory Overview	7
規劃檔案系統	8
定義檔案和區塊節點	10
定義BeeGFS服務	25
將BeeGFS服務對應至檔案節點	31
部署BeeGFS檔案系統	32
Ansible教戰手冊總覽	32
部署BeeGFS HA叢集	33
部署BeeGFS用戶端	36
驗證BeeGFS部署	41

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。

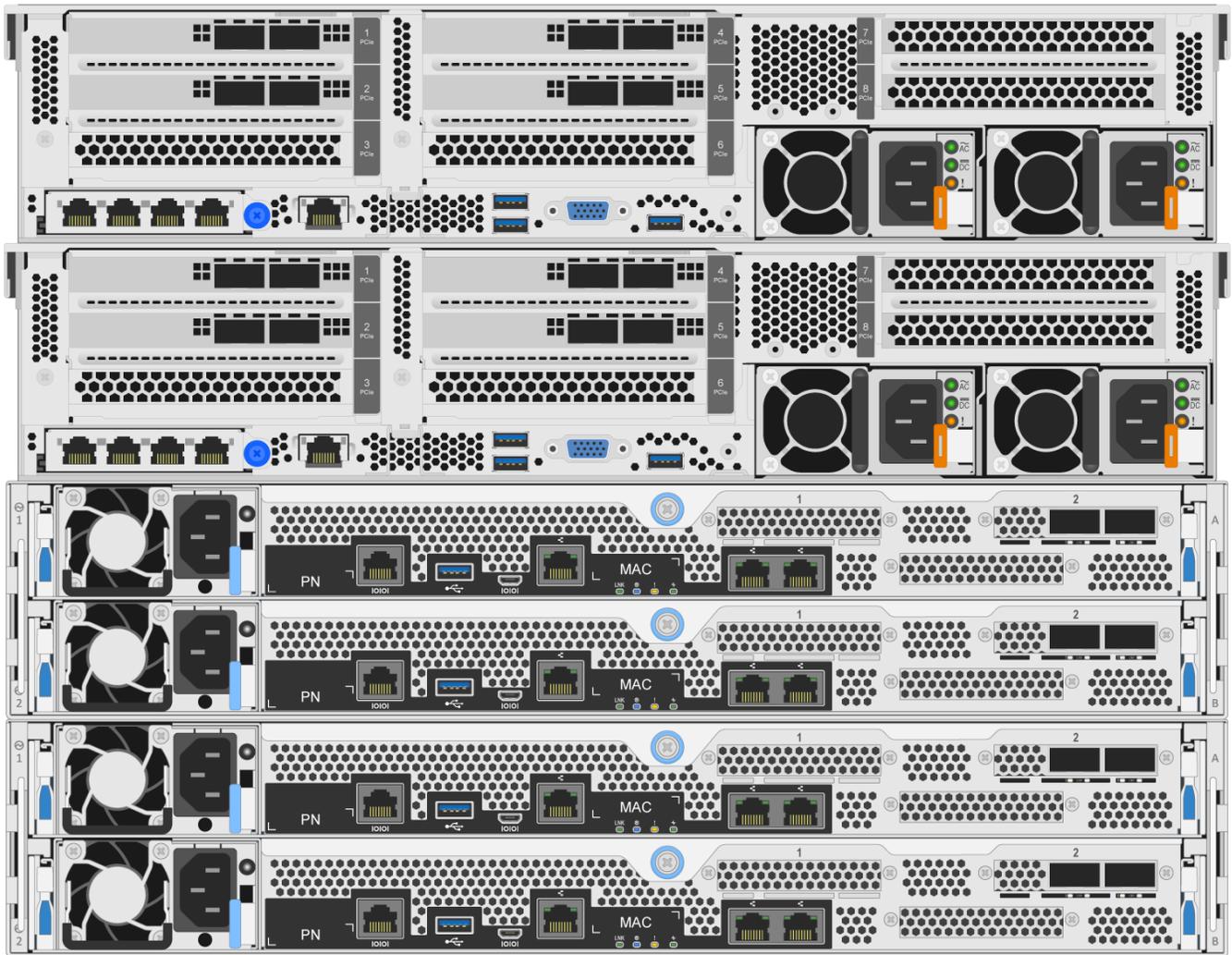


在繼續安裝之前、請先確認您已充分瞭解特定BeeGFS部署所需的檔案系統架構。

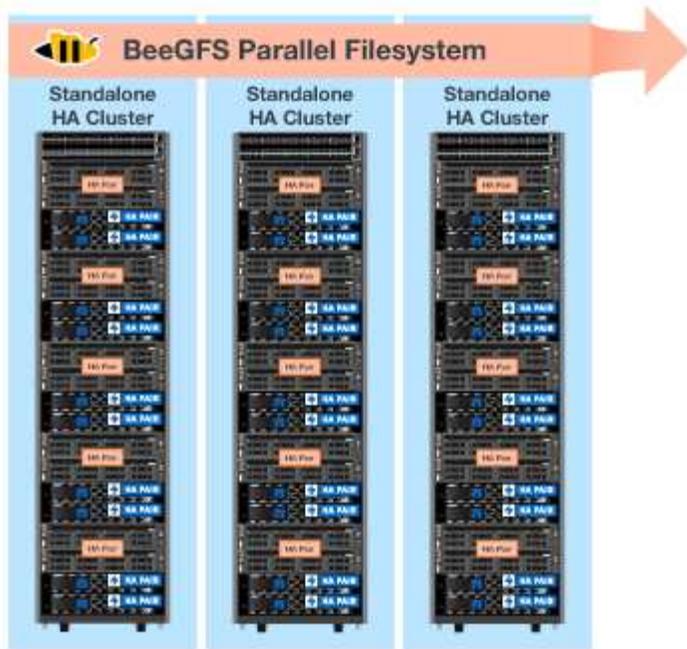
機架硬體

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：

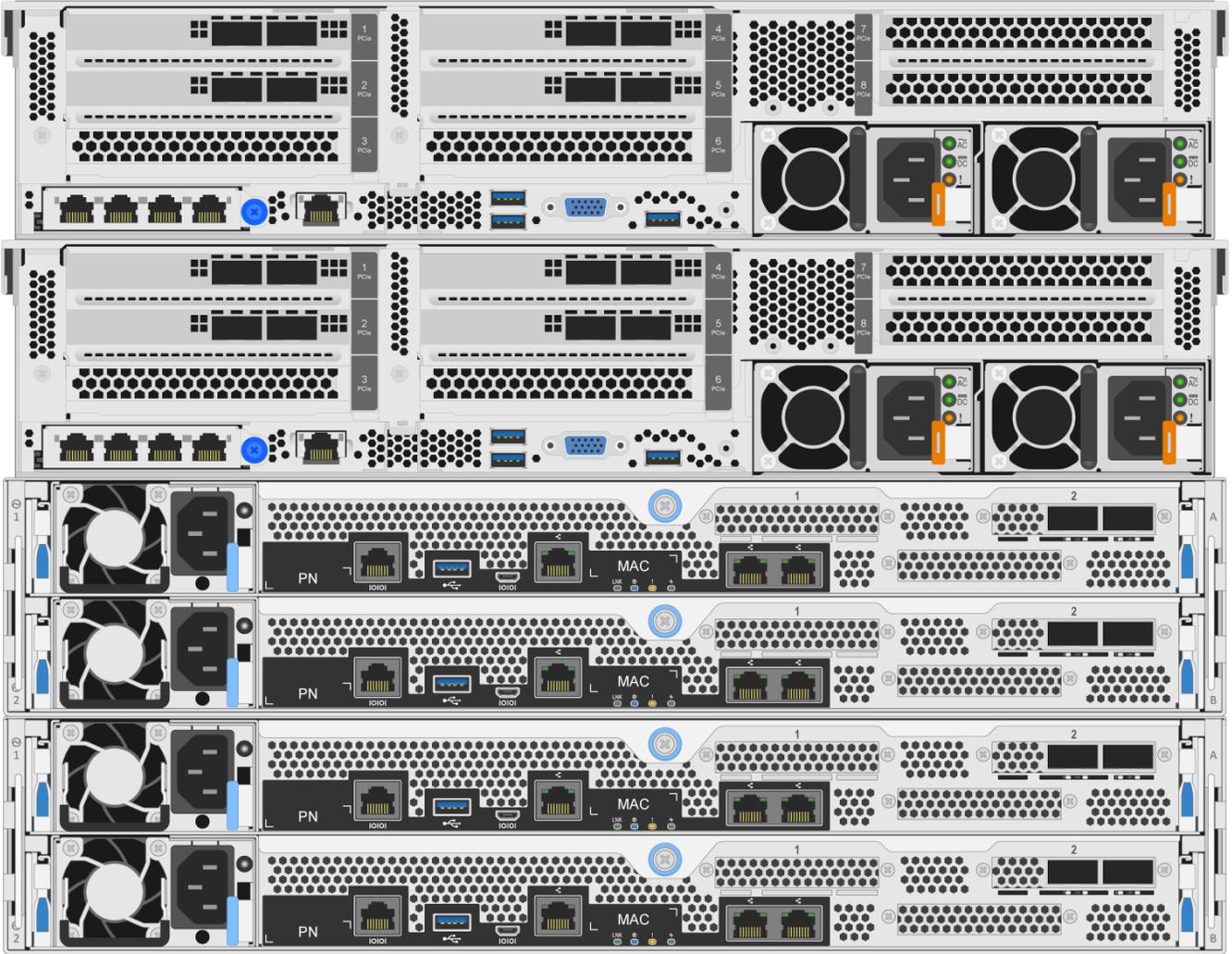


大型BeeGFS安裝範例、其中每個HA叢集有多個建置區塊、以及檔案系統中的多個HA叢集：



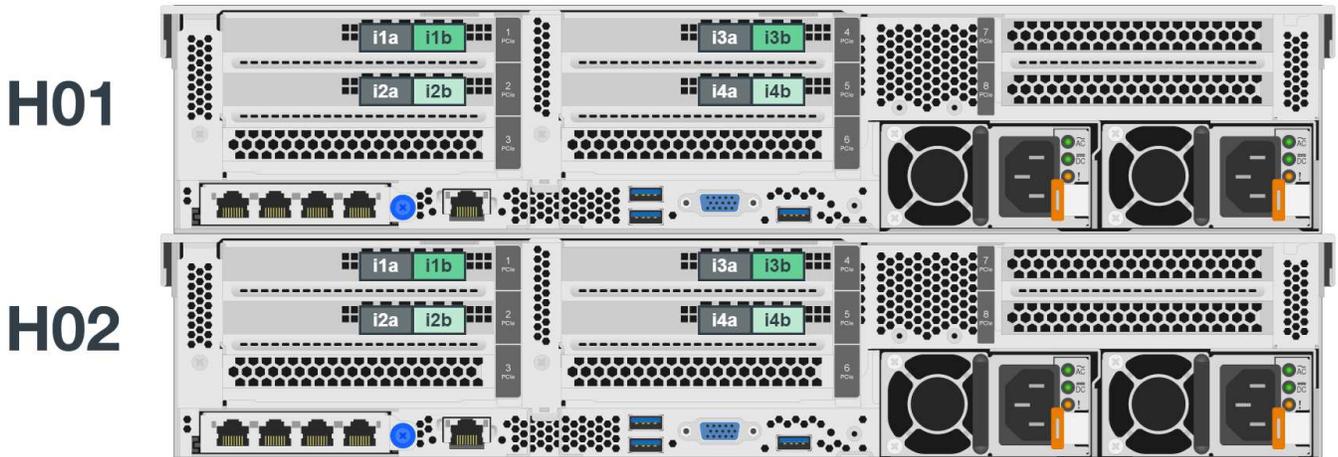
纜線檔案和區塊節點

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

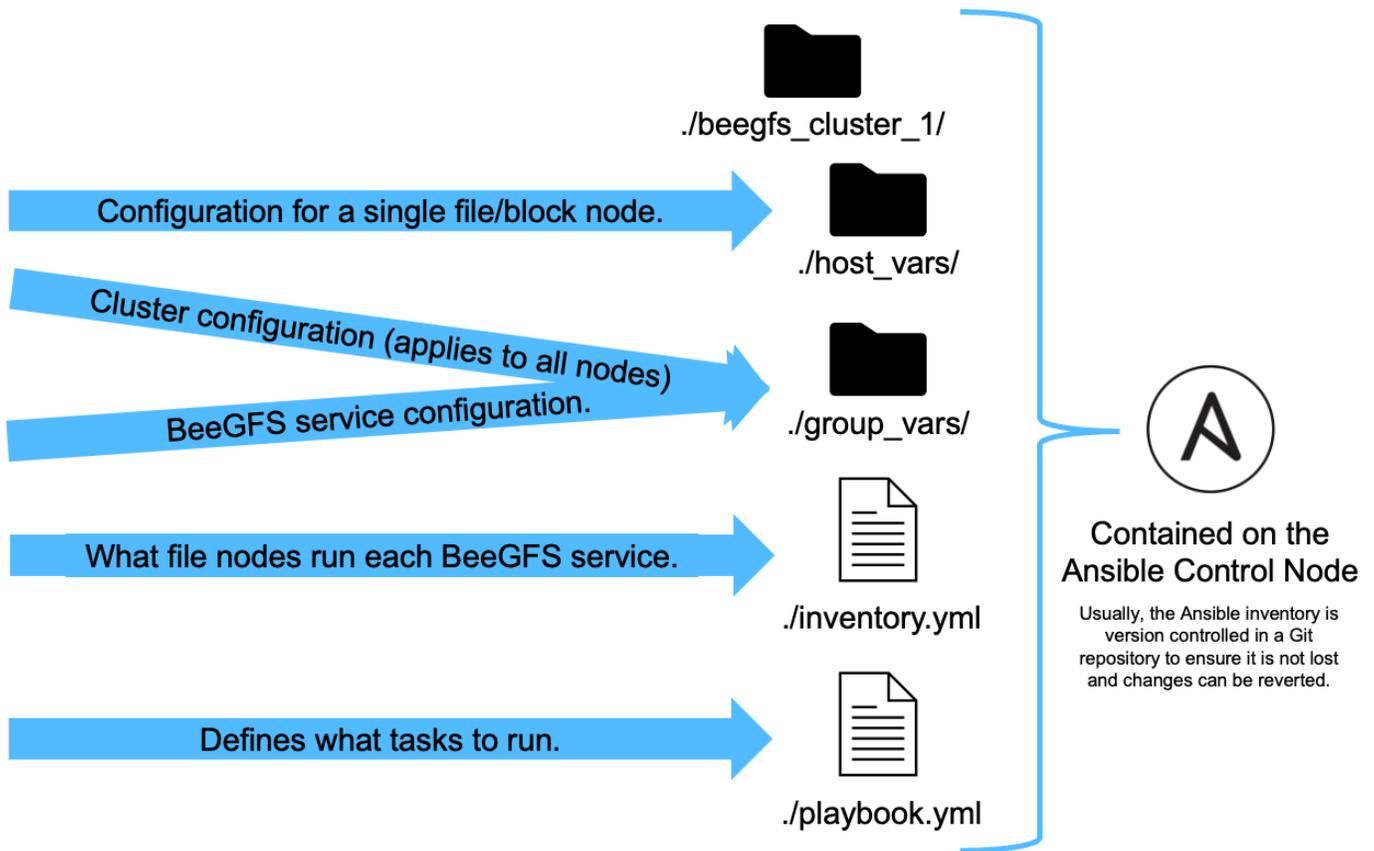
Ansible Inventory Overview

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>.yaml` 或 `storage_<ID>.yaml` 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 (起搏器和電量器同步) 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 ilb
  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 `e-series-`

[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>.yaml` 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>/yaml`、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_mgmt_d_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_mgmt_d_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: ib1a
```

- 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_mgmt_d_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/<sysMgmtHost>_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_nvram: "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-mgmt.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：`storeMgmtDirectory`、`connAuthFile`、`connDisableAuthentication`、`connInterfacesFile` 和 `connNetFilterFile`。

```
beegfs_ha_beegfs_mgmt_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：
`sysMgmdHost`、`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：`sysMgmtHost`、`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*幂等*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_beeegfs
```

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/<sysMgmtHost>_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、以使用戶端連線至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 ilb
  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:
      - sysMgmtHost: <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/<sysMgmtHost>_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。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。