使用自定义架构

BeeGFS on NetApp with E-Series Storage

NetApp
December 01, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/beegfs/custom/architectures-overview.html on December 01, 2025. Always check docs.netapp.com for the latest.

目录

使用自定义架构 1
概述和要求 1
简介 1
部署概述 1
要求 1
初始设置 2
安装硬件并为其布线 2
设置文件和块节点 6
设置Ansible控制节点 7
定义BeeGFS文件系统 7
Ansible清单概述 7
规划文件系统 8
定义文件和块节点 10
定义BeeGFS服务 25
将BeeGFS服务映射到文件节点 31
部署BeeGFS文件系统 32
Ansible攻略手册概述 32
部署BeeGFS HA集群 33
部署BeeGFS客户端 36
验证BeeGFS部署 41

使用自定义架构

概述和要求

使用Ansible部署BeeGFS高可用性集群时、请使用任何NetApp E/EF系列存储系统作为BeeGFS块节点、使用x86服务器作为BeeGFS文件节点。

本节中使用的术语定义可在"术语和概念"页面上找到。

简介

虽然"经过NetApp验证的架构"提供了预定义的参考配置和规模估算指导、但某些客户和合作伙伴可能更愿意设计更适合特定要求或硬件首选项的自定义架构。在NetApp上选择BeeGFS的主要优势之一是、能够使用Ansible部署BeeGFS共享磁盘HA集群、从而简化集群管理、并通过NetApp编写的HA组件提高可靠性。在NetApp上部署自定义BeeGFS架构仍可使用Ansible来完成、从而在一系列灵活的硬件上保持类似设备的方法。

本节概述了在NetApp硬件上部署BeeGFS文件系统以及使用Ansible配置BeeGFS文件系统所需的常规步骤。有关设计BeeGFS文件系统的最佳实践和优化示例的详细信息、请参见"经过NetApp验证的架构"一节。

部署概述

通常、部署BeeGFS文件系统涉及以下步骤：

- 初始设置：
  - 安装硬件/为硬件布线。
  - 设置文件和块节点。
  - 设置Ansible控制节点。
- 将BeeGFS文件系统定义为Ansible清单。
- 对文件和块节点运行Ansible以部署BeeGFS。
  - 也可以设置客户端并挂载BeeGFS。

后续章节将更详细地介绍这些步骤。

Ansible负责处理所有软件配置和配置任务、包括：

- 在块节点上创建/映射卷。
- 格式化/调整文件节点上的卷。
- 在文件节点上安装/配置软件。
- 建立HA集群并配置BeeGFS资源和文件系统服务。

要求

在Ansible中支持BeeGFS的发布时间为 "Ansible Galax河" 作为一组角色和模块、用于自动执行BeeGFS HA集群

的端到端部署和管理。

BeeGFS本身采用<major> <major>。<minor>。<patch> 版本控制方案进行版本控制、该集合会为每个受支持的BeeGFS的BeeGFS <minor> 版本维护角色、例如BeeGFS 7.2或BeeGFS 7.3。发布此集合的更新后、每个角色中的修补程序版本将更新为指向该版本分支的最新可用BeeGFS版本(例如：7.2.8)。该集合的每个版本也都经过测试并支持特定的 Linux 发行版和版本，目前文件节点使用 Red Hat，客户端使用 Red Hat 和 Ubuntu。不支持运行其他分发版、也不建议运行其他版本(尤其是其他主要版本)。

Ansible控制节点

此节点将包含用于管理BeeGFS的清单和攻略手册。它需要：

- Ansible 6.x (Ansible核心2.13)
- Python 3.6 (或更高版本)
- Python (pip)软件包：ipaddr和netaddr

此外、建议您将控制节点中的无密码SSH设置为连接到所有BeeGFS文件节点和客户端。

BeeGFS文件节点

文件节点必须运行 Red Hat Enterprise Linux (RHEL) 9.4，并有权访问包含所需软件包（pacemaker、corosync、fence-agents-all、resource-agents）的 HA 存储库。例如，可以执行以下命令在 RHEL 9 上启用相应的存储库：

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

BeeGFS客户端节点

可以使用BeeGFS客户端Ansible角色安装BeeGFS客户端软件包并管理BeeGFS挂载。此角色已使用 RHEL 9.4 和 Ubuntu 22.04 进行测试。

如果未使用Ansible设置BeeGFS客户端并挂载BeeGFS、则为any "BeeGFS支持的Linux分发版和内核" 可以使用。

初始设置

安装硬件并为其布线

安装用于在NetApp上运行BeeGFS的硬件并为其布线所需的步骤。

规划安装

每个BeeGFS文件系统将包含一些文件节点、这些文件节点运行BeeGFS服务、并使用由某些数量的块节点提供的后端存储。文件节点配置为一个或多个高可用性集群、以便为BeeGFS服务提供容错功能。每个块节点都已是一个主动/主动HA对。每个HA集群中支持的文件节点的最小数量为3个、每个集群中支持的文件节点的最大数量为10个。BeeGFS文件系统可以通过部署多个独立的HA集群来扩展到10个以上节点、这些集群可以协同工作来提供一个文件系统命名空间。

通常、每个HA集群都部署为一系列"组件"、其中、一些文件节点(x86服务器)直接连接到某些数量的块节点(通常为E系列存储系统)。此配置会创建一个非对称集群、在此集群中、BeeGFS服务只能在某些文件节点上运行、这些文件节点可以访问用于BeeGFS目标的后端块存储。每个组件中的文件到块节点与直接连接所使用的存储协议的平衡取决于特定安装的要求。

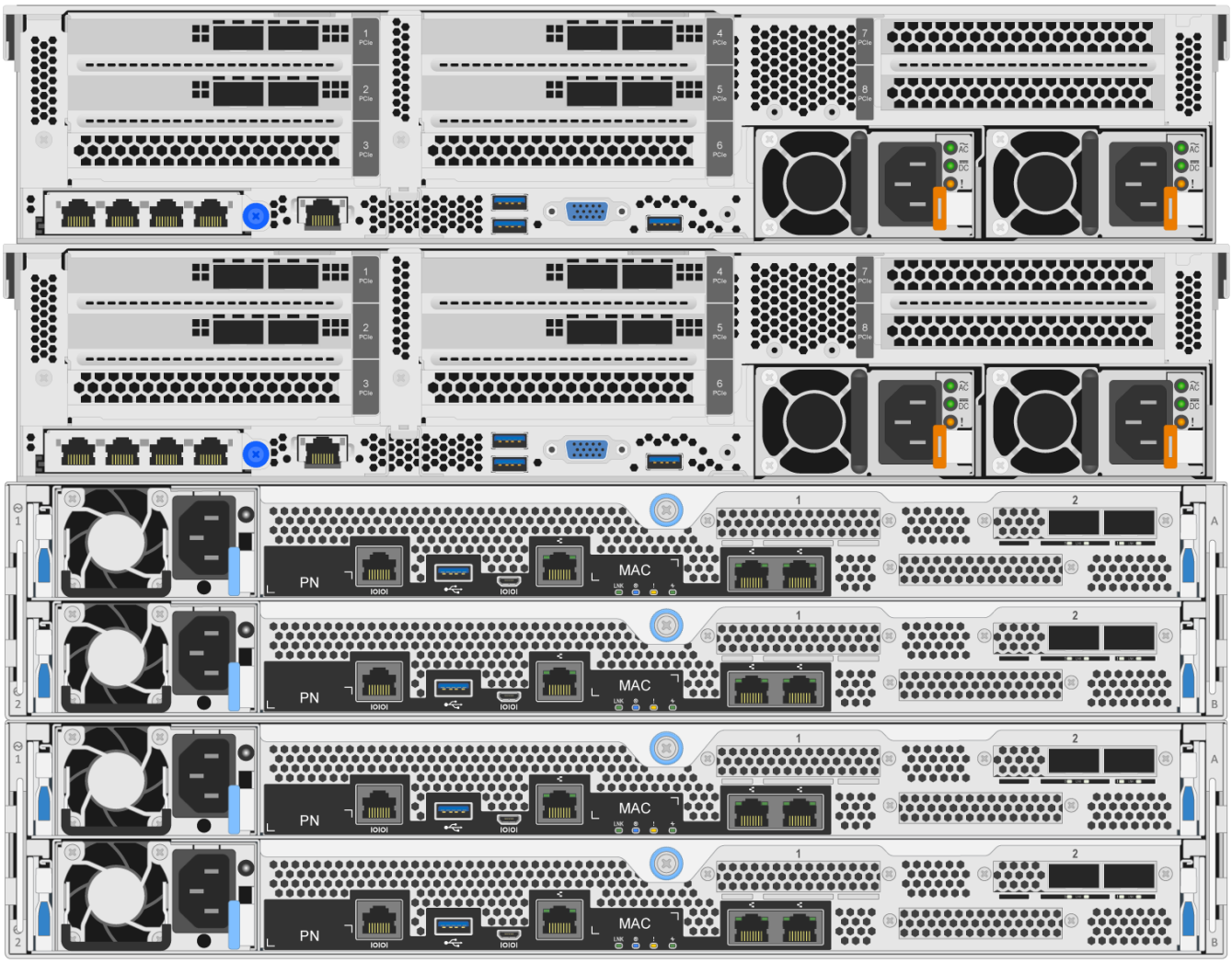
另一种HA集群架构使用文件节点和块节点之间的存储网络结构(也称为存储区域网络或SAN)来建立对称集群。这样、BeeGFS服务便可在特定HA集群中的任何文件节点上运行。由于增加了SAN硬件、一般对称集群的成本效益不高、因此本文档假定使用的是部署为一系列一个或多个组件的非对称集群。

> 在继续安装之前、请确保已充分了解特定BeeGFS部署所需的文件系统架构。

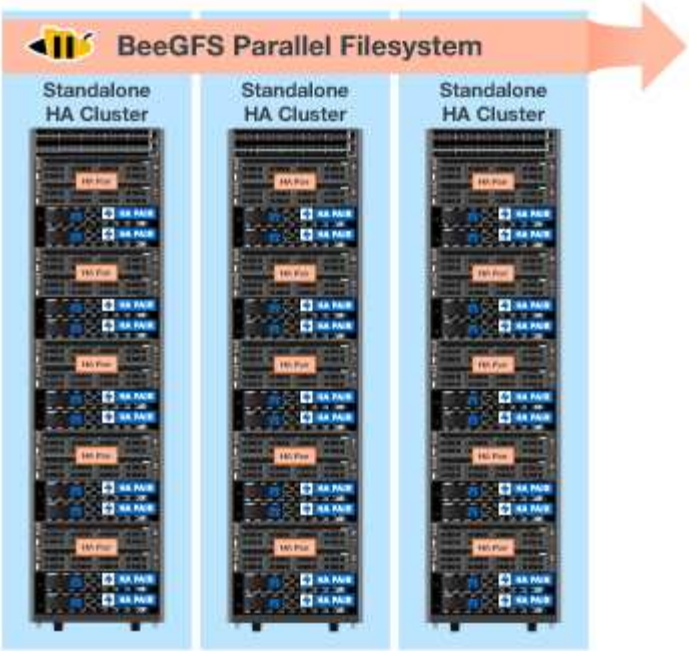
机架硬件

在规划安装时、每个组件中的所有设备都必须安装在相邻的机架单元中。最佳实践是、将文件节点直接机架安装在每个构建块中的块节点上方。按照文件和型号对应的文档进行操作 "块" 在机架中安装导轨和硬件时所使用的节点。

单个组件示例：

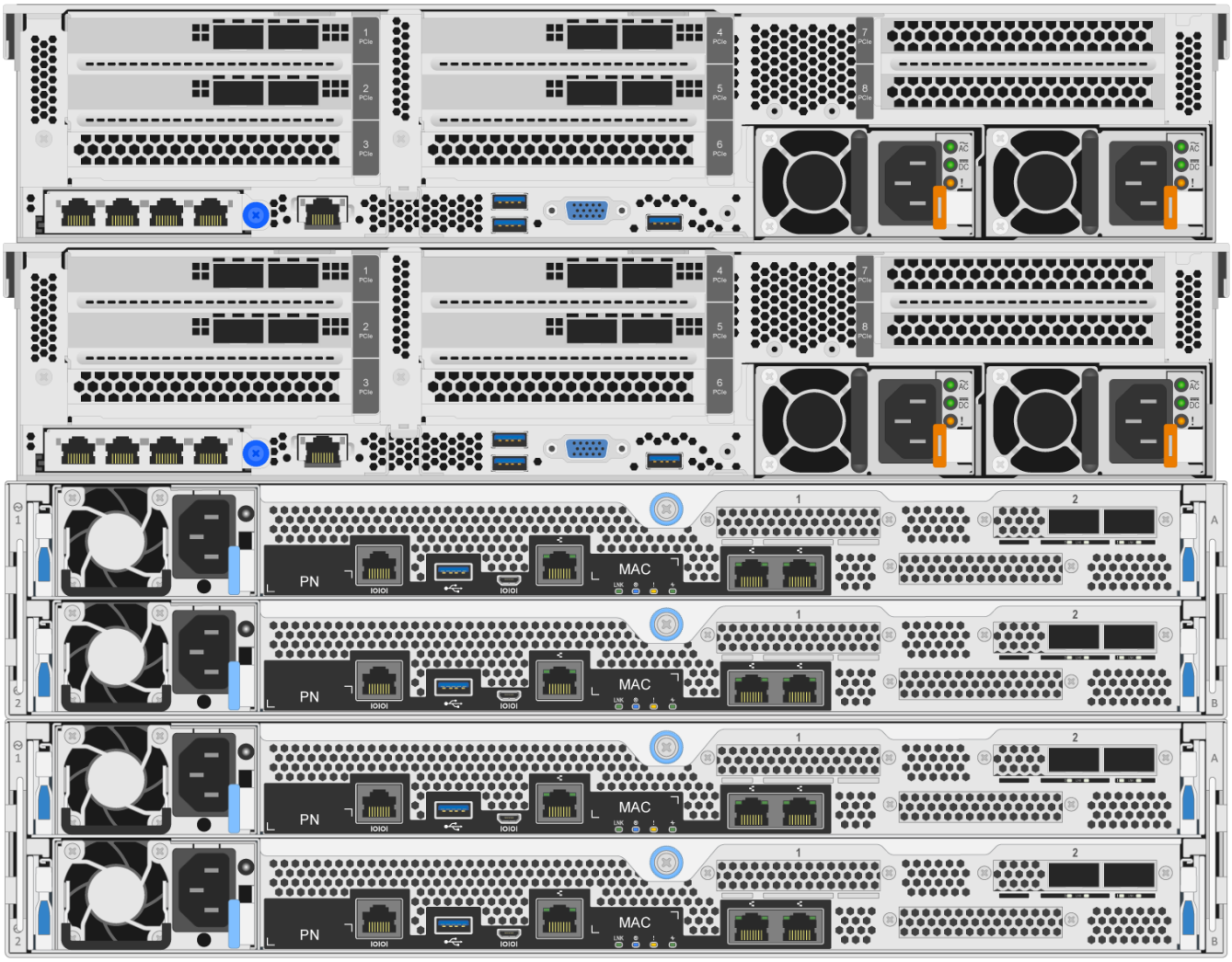


大型BeeGFS安装示例、其中每个HA集群中有多个组件、文件系统中有多个HA集群：



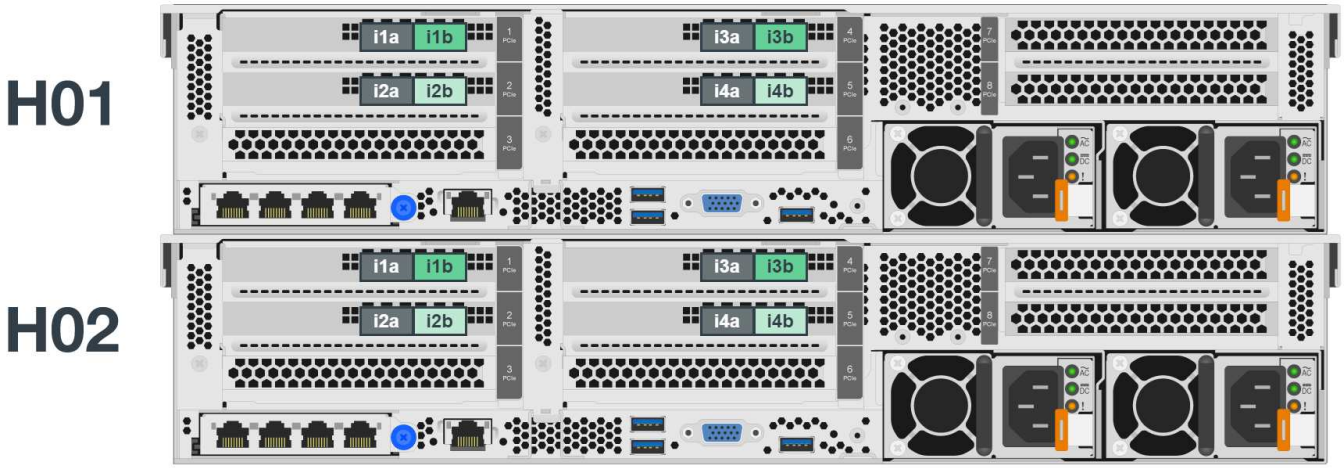
为文件和块节点布线

通常、您会将E系列块节点的HIC端口直接连接到文件节点的指定主机通道适配器(对于InfiniBand协议)或主机总线适配器(对于光纤通道和其他协议)端口。建立这些连接的确切方法取决于所需的文件系统架构，下面是一个示例"基于经过NetApp验证的第二代BeeGFS架构"：



使用缆线将文件节点连接到客户端网络

每个文件节点都将为BeeGFS客户端流量指定一定数量的InfiniBand或以太网端口。根据架构的不同、每个文件节点将有一个或多个连接到高性能客户端/存储网络、也可能连接到多个交换机以实现冗余并增加带宽。以下是使用冗余网络交换机进行客户端布线的示例、其中以深绿色和浅绿色突出显示的端口连接到不同的交换机：



连接管理网络和电源

建立带内和带外网络所需的任何网络连接。

连接所有电源、确保每个文件和块节点都连接到多个配电单元以实现冗余(如果可用)。

设置文件和块节点

在运行Ansible之前设置文件和块节点所需的手动步骤。

文件节点

配置基板管理控制器(BMC)

基板管理控制器(Baseboard Management Controller、BMC)有时也称为服务处理器、它是内置于各种服务器平台中的带外管理功能的通用名称、即使操作系统未安装或无法访问、也可以提供远程访问。供应商通常会使用自己的品牌推广此功能。例如、在联想SR665上、BMC称为联想XicLityController (XCC)。

按照服务器供应商的文档启用访问此功能所需的任何许可证、并确保BMC已连接到网络并已正确配置用于远程访问。

> 如果需要使用Redfish进行基于BMC的隔离、请确保已启用Redfish、并且可以从文件节点上安装的操作系统访问BMC界面。如果BMC和操作系统共享同一物理网络接口、则可能需要对网络交换机进行特殊配置。

调整系统设置

使用系统设置(BIOS/UEFI)界面、确保设置为最大程度地提高性能。确切设置和最佳值将因所使用的服务器型号而异。提供了有关的指导"已验证文件节点型号"，否则请参阅服务器供应商的文档和基于您的型号的最佳实践。

安装操作系统

根据列出的文件节点要求安装受支持的操作系统"此处"。根据您的Linux版本、请参见以下任何其他步骤。

Red Hat

使用 Red Hat Subscription Manager 注册并订阅系统，以允许从官方 Red Hat 存储库安装所需的软件包，并将更新限制在受支持的 Red Hat 版本上： `subscription-manager release --set=<MAJOR_VERSION>.<MINOR_VERSION>` 。有关说明，请参阅 "如何注册和订阅RHEL系统"和 "如何限制更新" 。

启用包含高可用性所需软件包的Red Hat存储库：

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

配置管理网络

配置所需的任何网络接口、以便对操作系统进行带内管理。具体步骤取决于所使用的特定Linux版本。

> 确保已启用SSH、并且可从Ansible控制节点访问所有管理接口。

更新**HCA**和**HBA**固件

确保所有HBA和HCA均运行上列出的受支持固件版本"NetApp 互操作性表"、并在必要时进行升级。有关NVIDIA ConnectX适配器的其他建议，请参见"此处"。

块节点

按照步骤执行操作 "启动并运行E系列" 在每个块节点控制器上配置管理端口、并可选择为每个系统设置存储阵列名称。

> 除了确保可从Ansible控制节点访问所有块节点之外、无需进行其他配置。其余系统配置将使用Ansible进行应用/维护。

设置**Ansible**控制节点

设置Ansible控制节点以部署和管理文件系统。

概述

Ansible控制节点是用于管理集群的物理或虚拟Linux计算机。它必须满足以下要求：

- 满足"要求"BeeGFS HA角色的要求、包括已安装的Ansient、Python版本以及任何其他Python软件包。
- 与官方人员会面 "Ansible控制节点要求" 包括操作系统版本。
- 对所有文件和块节点具有SSH和HTTPS访问权限。

有关详细的安装步骤，请参见"此处"。

定义**BeeGFS**文件系统

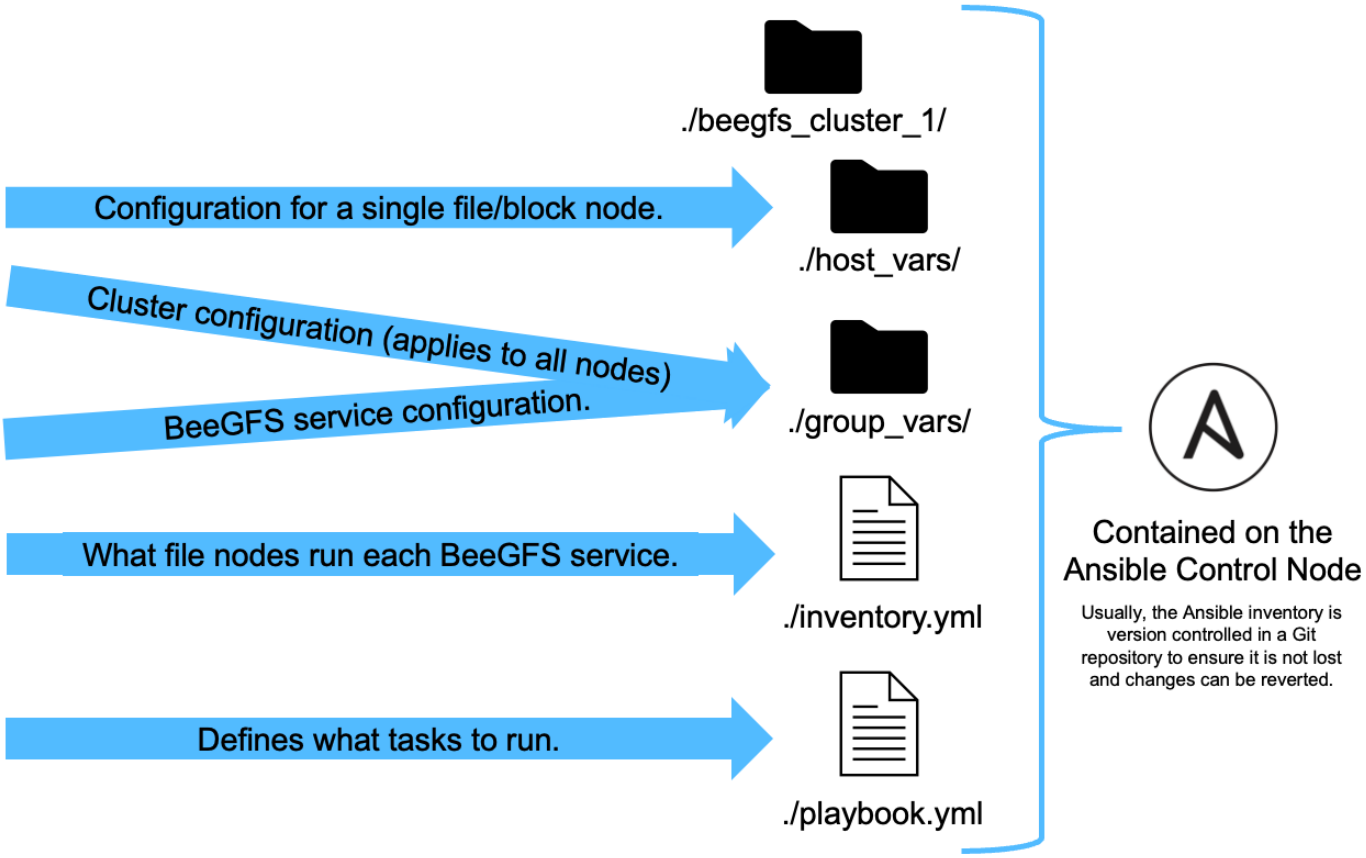
Ansible清单概述

Ansible清单是一组配置文件、用于定义所需的BeeGFS HA集群。

概述

建议遵循标准的Ansible实践来组织 "清单"、包括使用 "子目录/文件" 而不是将整个清单存储在一个文件中。

单个BeeGFS HA集群的Ansible清单组织如下：



由于一个BeeGFS文件系统可以跨越多个HA集群、因此大型安装可能具有多个Ansible清单。通常、不建议尝试将多个HA集群定义为一个Ansible清单以避免出现问题。

步骤

1. 在Ansible控制节点上、创建一个空目录、该目录将包含要部署的BeeGFS集群的Ansible清单。
   a. 如果您的文件系统最终将包含/可能包含多个HA集群、建议先为文件系统创建一个目录、然后为表示每个HA集群的清单创建子目录。例如：

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

2. 在包含要部署的HA集群清单的目录中、创建两个目录 `group_vars` 和 `host_vars` 和两个文件 `inventory.yml` 和 `playbook.yml`。

以下各节将逐步介绍如何定义每个文件的内容。

规划文件系统

在构建Ansible清单之前规划文件系统部署。

概述

在部署文件系统之前、您应定义集群中运行的所有文件节点、块节点和BeeGFS服务需要哪些IP地址、端口和其他配置。虽然具体配置会因集群架构而异、但本节定义了一般适用的最佳实践和应遵循的步骤。

步骤

1. 如果您使用基于IP的存储协议(例如iSER、iSCSI、NVMe/IB或NVMe/RoCE)将文件节点连接到块节点、请为每个组件填写以下工作表。一个构建块中的每个直接连接都应具有一个唯一的子网、并且不应与用于客户端-服务器连接的子网重叠。

文件节点	IB端口	IP 地址	块节点	IB端口	物理IP	虚拟IP (仅适用于具有HDR IB的EF600)
<HOSTNAME>	<PORT>	<IP/SUBNET>	<HOSTNAME>	<PORT>	<IP/SUBNET>	<IP/SUBNET>

> 如果每个构建块中的文件和块节点直接连接、您通常可以对多个构建块重复使用相同的IP/方案。

2. 无论您是在存储网络中使用InfiniBand还是基于融合以太网的RDMA (RoCE)、请填写以下工作表以确定将用于HA集群服务、BeeGFS文件服务和客户端进行通信的IP范围：

目的	**InfiniBand**端口	**IP**地址或范围
BeeGFS集群IP	<INTERFACE(s)>	<RANGE>
BeeGFS管理	<INTERFACE(s)>	<IP(s)>
BeeGFS元数据	<INTERFACE(s)>	<RANGE>
BeeGFS存储	<INTERFACE(s)>	<RANGE>
BeeGFS客户端	<INTERFACE(s)>	<RANGE>

   a. 如果您使用的是一个IP子网、则只需要一个工作表、否则还需要填写第二个子网的工作表。

3. 根据上述内容、为集群中的每个组件填写以下工作表、以定义要运行的BeeGFS服务。对于每个服务、请指定首选/二级文件节点、网络端口、浮动IP、NUMA分区分配(如果需要)以及要用于其目标的块节点。填写工作表时、请参见以下准则：

   a. 将BeeGFS服务指定为 `mgmt.yml`， `meta_<ID>.yml`或 `storage_<ID>.yml` 其中ID表示此文件系统中此类型的所有BeeGFS服务的唯一编号。此约定将简化在创建用于配置每个服务的文件时在后续章节中引用此工作表的过程。

   b. 用于BeeGFS服务的端口只需在特定组件中是唯一的。确保具有相同端口号的服务不能在同一文件节点上运行、以避免端口冲突。

   c. 如果需要、服务可以使用多个块节点和/或存储池中的卷(并非所有卷都需要归同一控制器所有)。多个服务也可以共享同一个块节点和/或存储池配置(各个卷将在后面的章节中进行定义)。

BeeGFS服务(文件名)	文件节点	Port	浮动IP	NUMA区域	块节点	存储池	所属控制器
<SERVICE TYPE> <ID>	<PREFERRED FILE NODE> <SECONDARY FILE NODE(s)>	<PORT>	<INTERFACE>：<IP/SUBNET> <INTERFACE>：<IP/SUBNET>	<NUMA NODE/ZONE>	<BLOCK NODE>	<STORAGE POOL/VOLUME GROUP>	<A OR B>

有关标准约定、最佳实践和填写的示例工作表的详细信息、请参阅"最佳实践""定义BeeGFS组件"NetApp经验证的架构上BeeGFS的和部分。

定义文件和块节点

配置单个文件节点

使用主机变量(host_vars)指定各个文件节点的配置。

概述

本节将逐步介绍填充 `host_vars/<FILE_NODE_HOSTNAME>.yml` 集群中每个文件节点的文件。这些文件只能包含特定文件节点独有的配置。这通常包括：

- 定义Ansible连接到节点时应使用的IP或主机名。
- 配置用于HA集群服务(Pacemaker和Corosync)的其他接口和集群IP、以便与其他文件节点进行通信。默认情况下、这些服务与管理接口使用相同的网络、但应提供更多接口以实现冗余。通常的做法是、在存储网络上定义其他IP、从而避免需要额外的集群或管理网络。
  - 用于集群通信的任何网络的性能对于文件系统性能并不重要。使用默认集群配置时、通常至少1 Gb/秒的网络可为集群操作(例如同步节点状态和协调集群资源状态更改)提供足够的性能。慢速/繁忙网络可能会使发生原因 资源状态发生变化、所需时间比平常要长、在极端情况下、如果节点无法在合理的时间范围内发送检测信号、可能会导致节点被逐出集群。
- 配置用于通过所需协议连接到块节点的接口(例如：iSCSI/iSER、NVMe/IB、NVMe/RoCE、FCP等)

步骤

请参考"规划文件系统"本节中定义的IP寻址方案、为集群中的每个文件节点创建一个文件`host_vars/<FILE_NODE_HOSTNAME>/yml`并按如下所示进行填充：

1. 在顶部、指定Ansible通过SSH连接到节点并对其进行管理时应使用的IP或主机名：

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

2. 配置可用于集群流量的其他IP：
   a. 网络类型为 "InfiniBand (使用IPoIB)"：

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

b. 网络类型为 "基于融合以太网的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. 网络类型为 "以太网(仅限TCP、无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. 指示哪些IP应用于集群流量、优先IP列在上面：

```
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.
```

> 步骤2中配置的IP不会用作集群IP、除非它们包含在中 `beegfs_ha_cluster_node_ips` 列表这样、您就可以使用Ansible配置其他IP /接口、如果需要、这些IP /接口可用于其他目的。

4. 如果文件节点需要通过基于IP的协议与块节点进行通信、则需要在相应的接口上配置IP、并安装/配置该协议所需的任何软件包。

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 "iSER"：

```
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. 其他协议：

i. 如果使用 "NVMe/FC"、不需要配置各个接口。BeeGFS集群部署将根据需要自动检测协议和安装/配置要求。如果您使用网络结构连接文件和块节点、请确保按照NetApp和交换机供应商的最佳实践正确对交换机进行分区。

ii. 使用FCP或SAS不需要安装或配置其他软件。如果使用FCP、请确保交换机已正确分区、如下所示 "NetApp" 以及交换机供应商的最佳实践。

iii. 目前不建议使用IB SRP。根据E系列块节点支持的内容、使用NVMe/IB或iSER。

单击 "此处" 有关表示单个文件节点的完整清单文件的示例。

高级：在以太网模式和**InfiniBand**模式之间切换**NVIDIA ConnectX VPI**适配器

NVIDIA ConnectX-Virtual Protocol Interconnect & reg；(VPI)适配器既支持InfiniBand、也支持以太网作为传输层。不会自动协商模式之间的切换，必须使用中包含的工具进行配置 `mstconfig` ，该工具 `mstflint`是的一个开放源代码软件包 "NVIDIA软件工具(MFT)"。只需更改一次适配器的模式即可。这可以手动完成、也可以作为使用清单部分配置的任何接口的一部分包含在Ands得以 自动检查/应用的清单中 `eseries-[ib|ib_iser|ipoib|nvme_ib|nvme_roce|roce]_interfaces:`。

例如、要将InfiniBand模式下的接口电流更改为以太网、以便用于RoCE：

1. 对于要配置的每个接口、请指定 `mstconfig` 作为指定的映射(或词典) `LINK_TYPE_P<N>` 其中： `<N>` 由接口的HCA端口号决定。。 `<N>` 可以通过运行来确定值 `grep PCI_SLOT_NAME`

`/sys/class/net/<INTERFACE_NAME>/device/uevent` 并从PCI插槽名称中将1添加到最后一个数字、然后转换为十进制值。

a. 例如给定 `PCI_SLOT_NAME=0000:2f:00.2` (2 + 1 → HCA端口3)→ `LINK_TYPE_P3: eth`：

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

有关更多详细信息、请参见 "NetApp E系列主机集合的文档" 所使用的接口类型/协议。

配置单个块节点

使用主机变量(host_vars)指定各个块节点的配置。

概述

本节将逐步介绍填充 `host_vars/<BLOCK_NODE_HOSTNAME>.yml` 集群中每个块节点的文件。这些文件只能包含特定块节点特有的配置。这通常包括：

- 系统名称(如System Manager中所示)。
- 其中一个控制器的HTTPS URL (用于使用其REST API管理系统)。
- 用于连接到此块节点的存储协议文件节点。
- 配置主机接口卡(HIC)端口、例如IP地址(如果需要)。

步骤

请参考"规划文件系统"本节中定义的IP寻址方案、为集群中的每个块节点创建一个文件`host_vars/<BLOCK_NODE_HOSTNAME>/yml`并按如下所示进行填充：

1. 在顶部指定其中一个控制器的系统名称和HTTPS URL：

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

2. 选择 "协议" 文件节点将用于连接到此块节点：

   a. 支持的协议： `auto`， `iscsi`， `fc`， `sas`， `ib_srp`， `ib_iser`， `nvme_ib`， `nvme_fc`， `nvme_roce`。

```
eseries_initiator_protocol: <PROTOCOL>
```

3. 根据所使用的协议、HIC端口可能需要额外配置。如果需要、应定义HIC端口配置、以便每个控制器配置中的顶部条目与每个控制器上最左侧的物理端口相对应、底部端口对应最右侧的端口。所有端口都需要有效配置、即使它们当前未在使用中也是如此。

> 如果要将HDR (200 GB) InfiniBand或200 GB RoCE与EF600块节点结合使用、另请参见以下部分。

a. 对于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. 对于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. 对于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. 对于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协议不需要额外配置。不正确建议使用SRP。

有关配置HIC端口和主机协议的其他选项、包括配置iSCSI CHAP的功能、请参见 "文档。" 包含在SANtricity 集合中。注意在部署BeeGFS时、存储池、卷配置以及配置存储的其他方面将在其他位置进行配置、不应在此文件中定义。

单击 "此处" 表示单个块节点的完整清单文件示例。

将HDR (200 GB) InfiniBand或200 GB RoCE与NetApp EF600块节点结合使用：

要将HDR (200 GB) InfiniBand与EF600结合使用、必须为每个物理端口配置第二个"虚拟" IP。下面是配置配备双端口InfiniBand HDR HIC的EF600的正确方法示例：

```
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)
```

指定通用文件节点配置

使用组变量(group_vars)指定通用文件节点配置。

概述

应连接到所有文件节点的配置在中进行定义 `group_vars/ha_cluster.yml`。通常包括：

- 有关如何连接和登录到每个文件节点的详细信息。
- 通用网络配置。
- 是否允许自动重新启动。
- 应如何配置防火墙和SELinux状态。
- 集群配置、包括警报和隔离。
- 性能调整。
- 常见BeeGFS服务配置。

> 此外、还可以在各个文件节点上定义此文件中设置的选项、例如、如果使用的是混合硬件型号、或者每个节点的密码不同。单个文件节点上的配置将优先于此文件中的配置。

步骤

创建文件 `group_vars/ha_cluster.yml` 并按如下所示进行填充：

1. 指示Ansible Control节点应如何向远程主机进行身份验证：

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

> 尤其是在生产环境中、不要以纯文本格式存储密码。请改用Ansible Vault (请参见 "使用Ansible Vault加密内容")或 `--ask-become-pass` 运行攻略手册时的选项。如果 `ansible_ssh_user` 已为root用户、则可以选择省略 `ansible_become_password`。

2. 如果要在以太网或InfiniBand接口上配置静态IP (例如集群IP)、并且多个接口位于同一IP子网中(例如、ib0使用192.168.1.10/24、ib1使用192.168.1.11/24)、要使多主机支持正常工作、必须设置其他IP路由表和规则。只需按如下所示启用提供的网络接口配置挂钩：

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

3. 部署集群时、根据存储协议的不同、可能需要重新启动节点、以便于发现远程块设备(E系列卷)或应用配置的其他方面。默认情况下、节点将在重新启动之前进行提示、但您可以通过指定以下内容来允许节点自动重新启动：

```
eseries_common_allow_host_reboot: true
```

   a. 默认情况下、重新启动后、为确保块设备和其他服务准备就绪、Ansible将等待系统完成 `default.target` 将在继续部署之前访问。在某些使用NVMe/IB的情况下、此过程可能不够长、无法初始化、发现和连接到远程设备。这可能会导致自动部署过早继续并失败。为避免在使用NVMe/IB时出现这种情况、还应定义以下内容：

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

4. 要使BeeGFS和HA集群服务进行通信、需要使用多个防火墙端口。除非您要手动配置固件(不建议)、否则请指定以下内容以创建所需的防火墙区域并自动打开端口：

```
beegfs_ha_firewall_configure: True
```

5. 目前不支持SELinux、建议将状态设置为disabled以避免冲突(尤其是在使用RDMA时)。设置以下内容以确保SELinux已禁用：

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

6. 配置身份验证以使文件节点能够进行通信、并根据组织策略根据需要调整默认值：

```
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. 根据"规划文件系统"指定此文件系统的BeeGFS管理IP部分：

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

> 虽然`冗余、但当您将BeeGFS文件系统扩展到单个HA集群之外时、`beegfs_ha_mgmtd_floating_ip非常重要。部署后续HA集群时无需额外的BeeGFS管理服务、并指向第一个集群提供的管理服务。

8. 根据需要启用电子邮件警报：

```
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. 强烈建议启用隔离、否则、在主节点发生故障时、可能会阻止服务在二级节点上启动。
   a. 通过指定以下内容全局启用隔离：

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

i. 注意如果需要、也可以在此处指定任何受支持的 "集群属性" 。通常不需要调整这些值，因为BeeGFS HA角色附带了许多经过充分测试的 "默认值"。

b. 接下来、选择并配置隔离代理：

i. 选项1：要使用APC配电单元(PDU)启用隔离、请执行以下操作：

```
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. 选项2：要使用联想XCC (及其他BMC)提供的Redffish API启用隔离：

```
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. 有关配置其他隔离代理的详细信息，请参见 "Red Hat 文档"。

10. BeeGFS HA角色可以应用多种不同的调整参数来帮助进一步优化性能。其中包括优化内核内存利用率和块设备I/O等参数。根据对NetApp E-Series块节点的测试、此角色会附带一组合理的 "默认值" 、但默认情况下、除非您指定以下内容、否则不会应用这些功能：

```
beegfs_ha_enable_performance_tuning: True
```

a. 如果需要、还可以在此处指定对默认性能调整所做的任何更改。有关更多详细信息、请参见完整 "性能调整参数" 文档。

11. 为了确保用于BeeGFS服务的浮动IP地址(有时称为逻辑接口)可以在文件节点之间进行故障转移、所有网络接口的名称必须一致。默认情况下、网络接口名称由内核生成、即使在安装了相同PCIe插槽中的网络适配器的相同服务器型号上、也不能保证生成一致的名称。在部署设备之前创建清单并知道生成的接口名称时、这

一点也很有用。以确保设备名称一致、具体取决于服务器或的方框图 `lshw -class network -businfo` 输出中、按如下所示指定所需的PCIe地址到逻辑接口映射：

a. 对于InfiniBand (IPoIB)网络接口：

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

b. 对于以太网网络接口：

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

> 为避免重命名接口时发生冲突(防止重命名接口)、不应使用任何可能的默认名称、例如eth0、ens9f0、ib0或ibs4f0。一种常见的命名约定是、对以太网或InfiniBand使用"e"或"i"、后跟PCIe插槽编号和一个字母以指示端口。例如、插槽3中安装的InfiniBand适配器的第二个端口为：i3b。

> 如果您使用的是经验证的文件节点型号、请单击 "此处" PCIe地址到逻辑端口映射示例。

12. (可选)指定应应用于集群中所有BeeGFS服务的配置。可以找到默认配置值 "此处"，并在其他位置指定每项服务配置：

a. BeeGFS管理服务：

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

b. BeeGFS元数据服务：

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

c. BeeGFS存储服务：

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

13. 截至BeeGFS 7.2.7和7.3.1 "连接身份验证" 必须配置或显式禁用。使用基于Ansible的部署可以通过以下几种方式进行配置：

a. 默认情况下、部署将自动配置连接身份验证并生成 `connauthfile` 该文件将分发到所有文件节点、并与BeeGFS服务结合使用。此文件还将放置/维护在位于的Ansible控制节点上

`<INVENTORY>/files/beegfs/<sysMgmtdHost>_connAuthFile` 应(安全)维护的位置、以便在需要访问此文件系统的客户端中重复使用。

i. 要生成新密钥、请指定 `-e "beegfs_ha_conn_auth_force_new=True` 运行Ansible攻略手册时。请注意、如果出现、则会忽略此问题 `beegfs_ha_conn_auth_secret` 已定义。

ii. 有关高级选项，请参阅附带的完整默认值列表 "BeeGFS HA角色"。

b. 可以通过在中定义以下内容来使用自定义密钥 `ha_cluster.yml`：

```
beegfs_ha_conn_auth_secret: <SECRET>
```

c. 可以完全禁用连接身份验证(不建议使用)：

```
beegfs_ha_conn_auth_enabled: false
```

单击 "此处" 有关表示通用文件节点配置的完整清单文件的示例。

将HDR (200 GB) InfiniBand与NetApp EF600块节点结合使用：

要将HDR (200 GB) InfiniBand与EF600结合使用、子网管理器必须支持虚拟化。如果使用交换机连接文件和块节点、则需要在子网管理器管理器上为整个网络结构启用此功能。

如果块节点和文件节点使用InfiniBand直接连接、 `opensm`则必须在每个文件节点上为直接连接到块节点的每个接口配置一个实例。通过指定 `configure: true`时间来完成此"配置文件节点存储接口"操作。

目前、受支持的Linux分发版附带的收件箱版本 `opensm` 不支持虚拟化。而是需要从NVIDIA OpenFabFabric企业分发版(OFED)安装和配置版本 `opensm` 。尽管仍支持使用Ansible进行部署、但还需要执行一些额外步骤：

1. 使用cURL或所需工具、将NVIDIA网站上部分中列出的OpenSM版本的软件包下载到目录中 "技术要求" `<INVENTORY>/packages/` 。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. 下 `group_vars/ha_cluster.yml` 定义以下配置：

```
### 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"
```

指定通用块节点配置

使用组变量(group_vars)指定通用块节点配置。

概述

应通过定义适用于所有块节点的配置 `group_vars/eseries_storage_systems.yml`。通常包括：

- 有关Ansible控制节点应如何连接到用作块节点的E系列存储系统的详细信息。
- 节点应运行的固件、NVSRAM和驱动器固件版本。
- 全局配置、包括缓存设置、主机配置以及应如何配置卷的设置。

此外、还可以在各个块节点上定义此文件中设置的选项、例如、如果使用的是混合硬件型号、或者每个节点的密码不同。单个块节点上的配置将优先于此文件中的配置。

步骤

创建文件 `group_vars/eseries_storage_systems.yml` 并按如下所示进行填充：

1. Ansible不使用SSH连接到块节点、而是使用REST API。为此、我们必须设置：

```
ansible_connection: local
```

2. 指定用于管理每个节点的用户名和密码。用户名可以选择省略(并默认为admin)、否则您可以指定具有管理员权限的任何帐户。此外、还应指定是验证SSL证书还是忽略SSL证书：

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

不建议以纯文本格式列出任何密码。使用Ansible存储或提供 `eseries_system_password` 使用-extra vars运行Ansible时。

3. 也可以指定应在节点上安装的控制器固件、NVSRAM和驱动器固件。需要将这些文件下载到 `packages/` 目录、然后运行Ansible。可以下载E系列控制器固件和NVSRAM "此处" 和驱动器固件 "此处"：

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

如果指定了此配置、Ansible将自动更新所有固件、包括重新启动控制器(如有必要)、而不会出现其他提示。这对BeeGFS/主机I/O预期不会造成中断、但发生原因 性能可能会暂时下降。

4. 调整全局系统配置默认值。此处列出的选项和值通常建议用于NetApp上的BeeGFS、但可以根据需要进行调整：

```
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. 配置全局卷配置默认值。此处列出的选项和值通常建议用于NetApp上的BeeGFS、但可以根据需要进行调整：

```
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. 如果需要、请根据以下最佳实践调整Ansible为存储池和卷组选择驱动器的顺序：
   a. 列出应首先用于管理和/或元数据卷、最后用于存储卷的任何(可能较小)驱动器。
   b. 确保根据磁盘架/驱动器机箱型号在可用驱动器通道之间平衡驱动器选择顺序。例如、如果使用EF600且不进行扩展、则驱动器0-11位于驱动器通道1上、驱动器12-23位于驱动器通道上。因此、要选择一种平衡驱动器选择的策略 `disk shelf:drive` 99：0、99：23、99：1、99：22等如果存在多个机箱、则第一个数字表示驱动器架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"
```

单击 "此处" 有关表示通用块节点配置的完整清单文件的示例。

定义BeeGFS服务

定义BeeGFS管理服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS管理服务。对于特定文件系统、HA集群中只应存在一个此类型的服务。配置此服务包括定义：

- 服务类型(管理)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS管理目标)。

步骤

创建一个新文件 `group_vars/mgmt.yml`并引用该"规划文件系统"部分、按如下所示填充该文件：

1. 指示此文件表示BeeGFS管理服务的配置：

```
beegfs_service: management
```

2. 定义应仅应用于此BeeGFS服务的任何配置。管理服务通常不需要执行此操作、除非您需要启用配额、无论您的任何受支持的配置参数如何 `beegfs-mgmtd.conf` 可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定： `storeMgmtdDirectory`， `connAuthFile`， `connDisableAuthentication`， `connInterfacesFile`，和 `connNetFilterFile`。

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

3. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS `connInterfacesFile` 选项)：

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS `connNetFilterFile` 选项)：

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

5. 根据以下准则指定此服务将在其中存储数据的BeeGFS管理目标：
   a. 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 `name`， `raid_level`， `criteria_*`，和 `common_*` 每个服务的配置(为每个服务列出的卷应不同)。
   b. 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可"此处"了解更多详细信息)。
   c. 单击 "此处" 以获取可用于的配置选项的完整列表 `eseries_storage_pool_configuration`。请注

意一些选项、例如 `state`， `host`， `host_type`， `workload_name`，和 `workload_metadata` 和卷名称将自动生成、不应在此指定。

```
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
```

单击 "此处" 有关表示BeeGFS管理服务的完整清单文件的示例。

定义**BeeGFS**元数据服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS元数据服务。对于特定文件系统、HA集群中应至少存在一个此类型的服务。配置此服务包括定义：

- 服务类型(元数据)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS元数据目标)。

步骤

参考"规划文件系统"部分、 `group_vars/meta_<ID>.yml`为集群中的每个元数据服务创建一个位于的文件、然后按如下所示填充它们：

1. 指示此文件表示BeeGFS元数据服务的配置：

   ```
   beegfs_service: metadata
   ```

2. 定义应仅应用于此BeeGFS服务的任何配置。您必须至少指定所需的TCP和UDP端口、无论这些端口来自哪个受支持的配置参数 `beegfs-meta.conf` 也可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定： `sysMgmtdHost`， `storeMetaDirectory`， `connAuthFile`， `connDisableAuthentication`， `connInterfacesFile`，和 `connNetFilterFile`。

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

3. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS `connInterfacesFile` 选项)：

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS `connNetFilterFile` 选项)：

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

5. 根据以下准则指定此服务将在其中存储数据的BeeGFS元数据目标(此操作也会自动配置) `storeMetaDirectory` 选项)：
   a. 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 `name`， `raid_level`， `criteria_*`，和 `common_*` 每个服务的配置(为每个服务列出的卷应不同)。
   b. 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可"此处"了解更多详细信息)。
   c. 单击 "此处" 以获取可用于的配置选项的完整列表 `eseries_storage_pool_configuration`。请注意一些选项、例如 `state`， `host`， `host_type`， `workload_name`，和 `workload_metadata` 和卷名称将自动生成、不应在此指定。

```
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
```

单击 "此处" 有关表示BeeGFS元数据服务的完整清单文件的示例。

定义**BeeGFS**存储服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS存储服务。对于特定文件系统、HA集群中应至少存在一个此类型的服务。配置此服务包括定义：

- 服务类型(存储)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS存储目标)。

步骤

参考"规划文件系统"部分、 `group_vars/stor_<ID>.yml`为集群中的每个存储服务创建一个文件、然后按如下所示填充它们：

1. 指示此文件表示BeeGFS存储服务的配置：

   ```
   beegfs_service: storage
   ```

2. 定义应仅应用于此BeeGFS服务的任何配置。您必须至少指定所需的TCP和UDP端口、无论这些端口来自哪个受支持的配置参数 `beegfs-storage.conf` 也可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定： `sysMgmtdHost`， `storeStorageDirectory`， `connAuthFile`， `connDisableAuthentication`， `connInterfacesFile`，和 `connNetFilterFile`。

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

3. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS `connInterfacesFile` 选项)：

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS `connNetFilterFile` 选项)：

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

5. 根据以下准则指定此服务要存储数据的BeeGFS存储目标(此操作也会自动配置) `storeStorageDirectory` 选项)：
   a. 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 `name`， `raid_level`， `criteria_*`，和 `common_*` 每个服务的配置(为每个服务列出的卷应不同)。
   b. 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可"此处"了解更多详细信息)。
   c. 单击 "此处" 以获取可用于的配置选项的完整列表 `eseries_storage_pool_configuration`。请注意一些选项、例如 `state`， `host`， `host_type`， `workload_name`，和 `workload_metadata` 和卷名称将自动生成、不应在此指定。

```
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
```

单击 "此处" 有关表示BeeGFS存储服务的完整清单文件的示例。

将BeeGFS服务映射到文件节点

使用指定可以运行每个BeeGFS服务的文件节点 `inventory.yml` 文件

概述

本节将介绍如何创建 `inventory.yml` 文件其中包括列出所有块节点并指定可以运行每个BeeGFS服务的文件节点。

步骤

创建文件 `inventory.yml` 并按如下所示进行填充：

1. 从文件顶部、创建标准Ansible清单结构：

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

2. 创建一个包含参与此HA集群的所有块节点的组：

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

3. 创建一个组、其中包含集群中的所有BeeGFS服务以及要运行这些服务的文件节点：

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

4. 对于集群中的每个BeeGFS服务、定义应运行该服务的首选文件节点和任何二级文件节点：

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

单击 "此处" 有关完整清单文件的示例。

部署BeeGFS文件系统

Ansible攻略手册概述

使用Ansible部署和管理BeeGFS HA集群。

概述

前面几节介绍了构建表示BeeGFS HA集群的Ansible清单所需的步骤。本节介绍由NetApp开发的用于部署和管理集群的Ansible自动化功能。

Ansible：关键概念

在继续操作之前、熟悉几个关键的Ansible概念会很有帮助：

- 根据Ansible清单执行的任务在称为*攻略手册*的内容中进行了定义。
  - Ansible中的大多数任务都设计为*幂等*、这意味着可以多次运行这些任务、以验证所需的配置/状态是否仍然适用、而不会造成中断或进行不必要的更新。
- Ansible中最小的执行单位是*模块*。

    - 典型的攻略手册使用多个模块。
        - 示例：下载软件包、更新配置文件、启动/启用服务。
    - NetApp分发模块以自动执行NetApp E系列系统。
- 更好地将复杂的自动化作为一个角色进行打包。
    - 基本上是分发可重复使用的攻略手册的标准格式。
    - NetApp为Linux主机和BeeGFS文件系统分发角色。

BeeGFS HA Role for Ansible：关键概念

在NetApp上部署和管理每个版本的BeeGFS所需的所有自动化功能均作为Ansible角色打包、并作为的一部分进行分发 "适用于BeeGFS的NetApp E系列Ansible资料集"：

- 可以将此角色视为BeeGFS的*安装程序*和现代*部署/管理*引擎之间的某个位置。
    - 将现代基础架构应用为代码实践和理念、以简化任何规模的存储基础架构管理。
    - 与此"Kubespray"项目允许用户部署/维护整个Kubirnetes分发版以实现横向扩展计算基础架构的方式类似。
- 此角色是NetApp用于打包、分发和维护基于NetApp的BeeGFS解决方案的*软件定义*格式。
    - 无需分发整个Linux分发版或大型映像、即可努力打造"类似设备的"体验。
    - 包括NetApp编写的符合Open Cluster Framework (OCF)的集群资源代理、用于自定义BeeGFS目标、IP地址和监控、从而实现智能Pacemaker/BeeGFS集成。
- 此角色不仅仅是部署"自动化"、还用于管理整个文件系统生命周期、包括：
    - 应用按服务或集群范围的配置更改和更新。
    - 解决硬件问题后自动执行集群修复和恢复。
    - 通过对BeeGFS和NetApp卷进行广泛测试来设置默认值、简化性能调整。
    - 验证并更正配置偏差。

NetApp还为提供了Ansible角色 "BeeGFS客户端"、可选择用于安装BeeGFS并将文件系统挂载到计算/GPU/登录节点。

部署BeeGFS HA集群

使用攻略手册指定部署BeeGFS HA集群时应运行的任务。

概述

本节介绍如何组合用于在NetApp上部署/管理BeeGFS的标准攻略手册。

步骤

创建Ansible攻略手册

创建文件 `playbook.yml` 并按如下所示进行填充：

1. 首先定义一组任务(通常称为 "播放")、并且只能在NetApp E系列块节点上运行。我们会使用暂停任务在运行

安装之前进行提示(以避免意外运行攻略手册)、然后导入 `nar_santricity_management` 角色。此角色负责应用中定义的任何常规系统配置 `group_vars/eseries_storage_systems.yml` 或个人 `host_vars/<BLOCK NODE>.yml` 文件。

```
- 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. 定义对所有文件和块节点运行的播放：

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

3. 在此角色中、我们可以选择定义一组"预任务"、这些任务应在部署HA集群之前运行。这对于验证/安装Python等任何前提条件非常有用。我们还可以注入任何飞行前检查、例如验证是否支持提供的Ansible标记：

```
  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. 最后、此Play会为您要部署的BeeGFS版本导入BeeGFS HA角色：

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

> 每个受支持的BeeGFS主要版本都会保留一个BeeGFS HA角色。这样、用户可以选择何时升级主要/次要版本。目前(`beegfs_7_3`(`beegfs_7_2`支持BeeGFS 7.3.x或BeeGFS 7.2.x。默认情况下、这两个角色将在发布时部署最新的BeeGFS修补程序版本、但用户可以选择覆盖此修补程序、并根据需要部署最新的修补程序。"升级指南"有关详细信息、请参见最新版本。

5. 可选：如果要定义其他任务、请记住这些任务是否应定向到 `all` 主机(包括E系列存储系统)或仅文件节点。如果需要、可使用定义一个专门针对文件节点的新Play - `hosts: ha_cluster`。

单击 "此处" 有关完整攻略手册文件的示例。

安装NetApp Ansible Collections

Ansible和所有依赖关系的BeeGFS收集将在上进行维护 "Ansible Galax河"。在Ansible控制节点上、运行以下命令以安装最新版本：

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

虽然通常不建议这样做、但也可以安装此集合的特定版本：

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

运行攻略手册

位于包含的Ansible控制节点上的目录中 `inventory.yml` 和 `playbook.yml` 文件、请按如下所示运行攻略手册：

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

根据集群的大小、初始部署可能需要20分钟以上的时间。如果部署因任何原因失败、只需更正任何问题(例如布线不当、节点未启动等)、然后重新启动Ansible攻略手册即可。

指定时"通用文件节点配置"，如果选择默认选项让Ansound自动管理基于连接的身份验证，则 `connAuthFile` 可在 `<playbook_dir>/files/beegfs/<sysMgmtdHost>_connAuthFile`(默认情况下)中找到用作共享密钥的。任何需要访问文件系统的客户端都需要使用此共享密钥。如果使用配置客户端，则会自动处理此"BeeGFS客户端角色"问题。

部署BeeGFS客户端

也可以使用Ansible配置BeeGFS客户端并挂载文件系统。

概述

要访问BeeGFS文件系统、需要在需要挂载文件系统的每个节点上安装和配置BeeGFS客户端。本节介绍如何使用可用执行这些任务 "Ansible角色"。

步骤

创建客户端清单文件

1. 如果需要、请从Ansible控制节点向要配置为BeeGFS客户端的每个主机设置无密码SSH：

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

2. 下 `host_vars/`下、为名为的每个BeeGFS客户端创建一个文件 `<HOSTNAME>.yml` 使用以下内容、在占位符文本中填写适用于您环境的正确信息：

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

3. 如果要使用NetApp E系列主机集合的角色配置InfiniBand或以太网接口、以便客户端连接到BeeGFS文件节点、也可以包括以下选项之一：

   a. 网络类型为 "InfiniBand (使用IPoIB)"：

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

   b. 网络类型为 "基于融合以太网的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. 网络类型为 "以太网(仅限TCP、无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. 创建新文件 `client_inventory.yml` 并指定Ansible应用于连接到每个客户端的用户、Ansible应用于权限升级的密码(这需要 `ansible_ssh_user` 为root用户或具有sudo权限)：

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

> 请勿以纯文本格式存储密码。请改用Ansible Vault (请参见) "Ansible文档" 使用Ansible Vault对内容进行加密)或使用 `--ask-become-pass` 运行攻略手册时的选项。

5. 在中 `client_inventory.yml` 文件中、列出应在下配置为BeeGFS客户端的所有主机 `beegfs_clients` 组、然后参考实时注释、取消注释在系统上构建BeeGFS客户端内核模块所需的任何其他配置：

```
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驱动程序时、请确保beegfs_client_OFED_include_path指向适用于您的Linux安装的正确"header include path"。有关详细信息，请参见的BeeGFS文档 "RDMA支持"。

6. 在中 `client_inventory.yml` 文件中、列出要在先前定义的任何下挂载的BeeGFS文件系统 `vars`：

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

7. 自BeeGFS 7.2.7和7.3.1起"连接身份验证"、必须配置或显式禁用。根据您在指定时选择配置基于连接的身份验证"通用文件节点配置"的方式，您可能需要调整客户端配置：

   a. 默认情况下、HA集群部署将自动配置连接身份验证并生成 `connauthfile` 该位置将放置/维护在位于的Ansible控制节点上 `<INVENTORY>/files/beegfs/<sysMgmtdHost>_connAuthFile`。默认情况下、BeeGFS客户端角色设置为读取此文件或将其分发到中定义的客户端 `client_inventory.yml`、不需要执行其他操作。

      i. 有关高级选项、请参阅随附的默认值完整列表 "BeeGFS客户端角色"。

   b. 如果选择使用指定自定义密钥 `beegfs_ha_conn_auth_secret` 在中指定 `client_inventory.yml` 文件：

```
beegfs_ha_conn_auth_secret: <SECRET>
```

   c. 如果选择完全禁用基于连接的身份验证 `beegfs_ha_conn_auth_enabled`、在中指定 `client_inventory.yml` 文件：

```
beegfs_ha_conn_auth_enabled: false
```

有关支持的参数的完整列表和其他详细信息、请参见 "完整的BeeGFS客户端文档"。有关客户端清单的完整示例、请单击 "此处"。

创建BeeGFS客户端攻略手册文件

1. 创建新文件 `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. 可选：如果要使用NetApp E系列主机集合的角色配置客户端连接到BeeGFS文件系统的接口、请导入与要配置的接口类型对应的角色：

   a. 如果您使用的是使用InfiniBand (IPoIB)：

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

   b. 如果您使用的是基于融合以太网的RDMA (RoCE)：

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

   c. 如果您使用的是以太网(仅限TCP、无RDMA)：

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

3. 最后、导入BeeGFS客户端角色以安装客户端软件并设置文件系统挂载：

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

有关客户端攻略手册的完整示例、请单击 "此处"。

运行BeeGFS客户端攻略手册

要安装/构建客户端并挂载BeeGFS、请运行以下命令：

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

验证BeeGFS部署

在将系统投入生产之前、请验证文件系统部署。

概述

在将BeeGFS文件系统投入生产之前、请执行一些验证检查。

步骤

1. 登录到任何客户端并运行以下命令、以确保所有预期节点均存在/可访问、并且未报告任何不一致或其他问题：

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

2. 关闭整个集群、然后重新启动它。从任何文件节点运行以下命令：

```
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. 将每个节点置于备用状态、并验证BeeGFS服务是否能够故障转移到二级节点。要登录到任何文件节点并运行以下命令：

```
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等性能基准测试工具验证文件系统性能是否满足预期。有关BeeGFS的常见测试和参数示例、请参见"设计验证"经验证的NetApp架构上的BeeGFS一节。

应根据为特定站点/安装定义的验收标准执行其他测试。

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。