



管理BeeGFS集群

BeeGFS on NetApp with E-Series Storage

NetApp

January 27, 2026

目录

管理BeeGFS集群	1
概述、关键概念和术语	1
概述	1
关键概念	1
通用术语	1
何时使用Ansible与pcs工具	2
检查集群的状态	2
概述	2
了解输出 <code>pcs status</code>	2
重新配置HA集群和BeeGFS	4
概述	4
如何禁用和启用隔离	4
更新HA集群组件	5
升级 BeeGFS 服务	5
升级到 BeeGFS v8	7
升级HA集群中的PacMaker和Corosync软件包	17
更新文件节点适配器固件	20
升级E-Series存储阵列	24
维护和维修	26
故障转移和故障恢复服务	26
将集群置于维护模式	28
停止并启动集群	29
替换文件节点	30
扩展或缩减集群	31
故障排除	32
概述	32
故障排除指南	32
常见问题	36
常见故障排除任务	37

管理BeeGFS集群

概述、关键概念和术语

了解如何在部署BeeGFS HA集群后对其进行管理。

概述

本节面向需要在部署BeeGFS HA集群后对其进行管理的集群管理员。即使是熟悉Linux HA集群的用户也应阅读本指南、因为在管理集群方面存在许多差异、尤其是在使用Ansible进行重新配置方面。

关键概念

虽然其中一些概念已在主["术语和概念"](#)页上介绍、但在BeeGFS HA集群环境中重新介绍它们会很有帮助：

****集群节点：***运行Pacemaker和Corosync服务并加入HA集群的服务器。

文件节点：用于运行一个或多个BeeGFS管理、元数据或存储服务的集群节点。

****块节点：***一种为文件节点提供块存储的NetApp E系列存储系统。这些节点不会加入BeeGFS HA集群、因为它们可以提供自己的独立HA功能。每个节点都包含两个存储控制器、这些控制器可在块层提供高可用性。

"BeeGFS服务：* BeeGFS管理、元数据或存储服务。每个文件节点将运行一个或多个服务、这些服务将使用块节点上的卷来存储其数据。

构建块： BeeGFS文件节点、E系列块节点以及在其上运行的BeeGFS服务的标准化部署、可简化按照NetApp经验证的架构横向扩展BeeGFS HA集群/文件系统的过程。自定义HA集群也受支持、但通常会采用类似的组件方法来简化扩展。

"BeeGFS HA Cluster"：*用于运行由块节点提供支持的BeeGFS服务以高可用性方式存储BeeGFS数据的文件节点数量可扩展。基于行业认可的开源组件Pacemaker和Corosync构建、并使用Ansible进行打包和部署。

****集群服务：****是指在加入集群的每个节点上运行的Pacemaker和Corosync服务。请注意、如果只需要两个文件节点、则节点可能不会运行任何BeeGFS服务、而是仅以"Tiebreaker "节点的形式加入集群。

****集群资源：****对于集群中运行的每个BeeGFS服务、您将看到一个BeeGFS监控资源和一个资源组、其中包含BeeGFS目标、IP地址(浮动IP)和BeeGFS服务本身的资源。

****可访问：***一种用于软件配置、配置管理和应用程序部署的工具、可将基础架构作为代码来启用。如何打包BeeGFS HA集群、以简化在NetApp上部署、重新配置和更新BeeGFS的过程。

**** pcs：**"" a command line interface available from any of the file nodes in the cluster used to query and control the state of nodes and resources in the cluster.

通用术语

****故障转移：***每个BeeGFS服务都有一个首选文件节点、除非该节点发生故障。当BeeGFS服务在非首选/二级文件节点上运行时、该服务称为处于故障转移状态。

****。故障恢复：***将BeeGFS服务从非首选文件节点移回其首选节点的操作。

"HA对："*可以访问同一组块节点的两个文件节点有时也称为HA对。这是整个NetApp通用的一个术语、指可以相互"接管"的两个存储控制器或节点。

维护模式：禁用所有资源监视，并阻止PacMaker移动或以其他方式管理群集中的资源(另请参见上的一节["维护模式"](#))。

- HA集群：*一个或多个运行BeeGFS服务的文件节点、这些文件节点可以在集群中的多个节点之间进行故障转移、以创建高可用性的BeeGFS文件系统。通常、文件节点会配置为HA对、这些HA对能够在集群中运行一部分BeeGFS服务。

何时使用Ansible与pcs工具

何时应使用Ansible与pcs命令行工具来管理HA集群？

所有集群部署和重新配置任务都应使用外部Ansible控制节点上的Ansible完成。集群状态的临时更改(例如、将节点置于待机状态和非待机状态)通常通过登录到集群的一个节点(最好是未降级或即将进行维护的节点)并使用pcs命令行工具来执行。

修改任何集群配置(包括资源、约束、属性和BeeGFS服务本身)都应始终使用Ansible完成。维护集群是为了维护Ansible清单和攻略手册的最新副本(理想情况下是为了控制源以跟踪更改)。如果需要更改配置、请更新清单并重新运行导入BeeGFS HA角色的Ansible攻略手册。

HA角色将处理将集群置于维护模式的问题、然后在重新启动BeeGFS或集群服务以应用新配置之前进行任何必要的更改。由于在初始部署之外通常不需要完全重新启动节点、因此重新运行Ansible通常被视为"安全"操作步骤、但始终建议在维护时段或非工作时间重新启动任何BeeGFS服务。这些重新启动通常不应发生原因 应用程序错误、但可能会影响性能(某些应用程序可能比其他应用程序处理得更好)。

如果您希望将整个集群恢复为完全最佳状态、也可以选择重新运行Ansible、并且在某些情况下、与使用pcs相比、恢复集群状态可能更容易。尤其是在集群因某种原因关闭的紧急情况下、所有节点都备份后、重新运行Ansible可能会比尝试使用pcs更快、更可靠地恢复集群。

检查集群的状态

使用pcs查看集群的状态。

概述

正在运行 `pcs status` 从任何集群节点上、都是查看集群整体状态和每个资源(例如BeeGFS服务及其依赖关系)状态的最简单方法。本节将介绍的输出内容 `pcs status` 命令：

了解输出 `pcs status`

运行 `pcs status` 在启动了集群服务(Pacemaker和Corosync)的任何集群节点上。输出顶部将显示集群的摘要：

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

以下部分列出了集群中的节点：

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

这主要表示处于备用或脱机状态的任何节点。处于备用状态的节点仍在参与集群、但被标记为不符合运行资源的条件。处于脱机状态的节点表示该节点上未运行集群服务、原因可能是已手动停止或节点已重新启动/关闭。



节点首次启动时、集群服务将停止、需要手动启动、以避免意外将资源故障恢复到运行状况不正常的节点。

如果节点由于非管理原因(例如故障)处于备用或脱机状态、则会在节点状态旁边用圆括号显示其他文本。例如、如果禁用了隔离、并且某个资源遇到了您将看到的故障 Node <HOSTNAME>: standby (on-fail)。另一种可能的状态是 Node <HOSTNAME>: UNCLEAN (offline)、该节点会被简要视为已隔离节点、但如果隔离失败、则此隔离将持续存在、指示集群无法确认节点的状态(这可能会阻止其他节点上启动资源)。

下一节显示了集群中所有资源及其状态的列表：

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

与节点类似、如果资源存在任何问题、则会在资源状态旁边用圆括号显示其他文本。例如、如果Pacemaker请求停止资源、但此操作无法在分配的时间内完成、则Pacemaker将尝试隔离此节点。如果禁用了隔离或隔离操作失败、则资源状态将为 FAILED <HOSTNAME> (blocked) Pacemaker将无法在其他节点上启动它。

值得注意的是、BeeGFS HA集群使用了许多经过BeeGFS优化的自定义OCF资源代理。特别是、BeeGFS监控器负责在特定节点上的BeeGFS资源不可用时触发故障转移。

重新配置HA集群和BeeGFS

使用Ansible重新配置集群。

概述

通常、要重新配置BeeGFS HA集群的任何方面、都应更新您的Ansient清单并重新运行`ansible-playbook`命令。其中包括更新警报、更改永久隔离配置或调整BeeGFS服务配置。这些`group\_vars/ha\_cluster.yml`选项可使用文件进行调整、可在["指定通用文件节点配置"](#)部分中找到完整的选项列表。

有关管理员在执行集群维护或维护时应注意的部分配置选项的其他详细信息、请参见下文。

如何禁用和启用隔离

设置集群时、默认情况下会启用/需要隔离。在某些情况下、可能需要暂时禁用隔离、以确保在执行某些维护操作(例如升级操作系统)时节点不会意外关闭。虽然可以手动禁用此功能、但管理员应注意一些利弊权衡。

选项1：使用**Ansible**禁用隔离(建议)。

使用Ansible禁用隔离后、BeeGFS监控器的"发生故障"操作将从"隔离"更改为"备用"。这意味着、如果BeeGFS监控器检测到故障、它将尝试将节点置于备用状态并对所有BeeGFS服务进行故障转移。在外部主动故障排除/测试中、这通常比选项2更理想。缺点是、如果某个资源无法在原始节点上停止、则会被阻止从其他位置启动(这就是生产集群通常需要隔离的原因)。

1. 位于Ansible清单中 `groups_vars/ha_cluster.yml` 添加以下配置：

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

2. 重新运行Ansible攻略手册、将所做的更改应用于集群。

选项2：手动禁用隔离。

在某些情况下、您可能希望在不重新运行Ansible的情况下临时禁用隔离、以便于对集群进行故障排除或测试。



在此配置中、如果BeeGFS监控器检测到故障、集群将尝试停止相应的资源组。它不会触发完全故障转移、也不会尝试重新启动受影响的资源组或将其移动到其他主机。要进行恢复、请解决所有问题、然后运行 `pcs resource cleanup` 或者手动将节点置于备用状态。

步骤

1. 要确定是否全局启用或禁用了隔离(stonith)、请运行：`pcs property show stonith-enabled`
2. 要禁用隔离、请运行：`pcs property set stonith-enabled=false`
3. 要启用隔离、请运行：`pcs property set stonith-enabled=true`



下次运行 Ansible 剧本时，此设置将被覆盖。

更新HA集群组件

升级 BeeGFS 服务

使用 Ansible 更新 HA 集群上运行的 BeeGFS 版本。

概述

BeeGFS采用 `major.minor.patch` 版本控制方案。每个受支持的 ``major.minor`` 版本 (例如 ``beegfs_ha_7_2`` 和 ``beegfs_ha_7_3``) 都提供了 BeeGFS HA Ansient 角色。每个 HA 角色都固定为在发布此 Ansient 集合时提供的最新 BeeGFS 修补程序版本。

所有 BeeGFS 升级都应使用 Ansible，包括在 BeeGFS 的主要、次要和补丁版本之间切换。要更新 BeeGFS，您首先需要更新 BeeGFS Ansient 集合，该集合还将引入部署/管理自动化和底层 HA 集群的最新修复和增强功能。即使更新到最新版本的集合，在运行 ``ansible-playbook`` 并设置 ``-e "beegfs_ha_force_upgrade=true"`` 之前，BeeGFS 也不会升级。有关每次升级的其他详细信息，请参阅当前版本的["BeeGFS 升级文档"](#)。



如果要升级到 BeeGFS v8，请参见["升级到 BeeGFS v8"](#)过程。

已测试升级路径

已测试和验证以下升级路径：

原始版本	升级版本	多导轨	详细信息
7.2.6	7.3.2	是的。	将beegfs收集从v3.0.1升级到v3.1.0、添加了多轨
7.2.6	7.2.8	否	将beegfs收集从v3.0.1升级到v3.1.0
7.2.8	7.3.1	是的。	使用beegfs集合v3.1.0进行升级、并添加了多轨
7.3.1	7.3.2	是的。	使用beegfs Collection v3.1.0进行升级
7.3.2	7.4.1	是的。	使用beegfs Collection v3.2.0进行升级
7.4.1	7.4.2	是的。	使用beegfs Collection v3.2.0进行升级
7.4.2	7.4.6	是的。	使用beegfs Collection v3.2.0进行升级
7.4.6	8.0	是的。	使用 "升级到 BeeGFS v8" 过程中的说明进行升级。
7.4.6	8.1	是的。	使用 "升级到 BeeGFS v8" 过程中的说明进行升级。
7.4.6	8.2	是的。	使用 "升级到 BeeGFS v8" 过程中的说明进行升级。

BeeGFS升级步骤

以下各节介绍了更新 BeeGFS 的 Ansient 集合和 BeeGFS 本身的步骤。请特别注意更新 BeeGFS 主要或次要版本的任何额外步骤。

第1步：升级BeeGFS集合

用于升级可访问的集合 ["Ansible Galaxy河"](#)下，运行以下命令：

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

要进行脱机收集升级、请从下载此收集 ["Ansible Galaxy河"](#) 单击所需的 Install Version` 然后 Download tarball。将tarball传输到Ansible控制节点并运行以下命令。

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

请参见 ["正在安装收集"](#) 有关详细信息 ...

第2步：更新Ands处理 清单

对集群的 Ansible 清单文件进行任何必需的或所需的更新。请参阅以下 [\[版本升级说明\]](#) 部分，了解有关特定升级要求的详细信息。有关配置 BeeGFS HA 清单的一般信息，请参阅 ["Ansible清单概述"](#) 部分。

第3步：更新《Ands维修 手册》(仅在更新主要或次要版本时)

如果要在主要版本或次要版本之间切换、请在用于部署和维护集群的文件中 `playbook.yml` 更新角色名称、`beegfs_ha_<VERSION>` 以反映所需的版本。例如，如果您要部署BeeGFS 7.4，则该文件为 `beegfs_ha_7_4`：

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

有关此播放手册文件内容的更多详细信息、请参见["部署BeeGFS HA集群"](#)一节。

第4步：运行BeeGFS升级

要应用BeeGFS更新、请执行以下操作：

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

BeeGFS HA角色将在后台处理：

- 确保集群处于最佳状态、每个BeeGFS服务都位于其首选节点上。
- 将集群置于维护模式。
- 更新HA集群组件(如果需要)。
- 按照以下步骤逐个升级每个文件节点：
 - 将其置于备用状态并将其服务故障转移到二级节点。
 - 升级BeeGFS软件包。
 - 回退服务。
- 将集群移出维护模式。

版本升级说明

从BeeGFS 7.2.6或7.3.0版升级

对基于连接的身份验证进行的更改

BeeGFS 版本 7.3.2 及更高版本要求配置基于连接的身份验证。如果没有以下任何一项，服务将无法启动：

- 指定一个 `connAuthFile`，或
- 在服务的配置文件中设置 `connDisableAuthentication=true`。

强烈建议启用基于连接的身份验证以实现安全。有关更多信息，请参见 ["基于BeeGFS连接的身份验证"](#)。

``beegfs_ha*`` 角色会自动生成身份验证文件并将其分发到：

- 集群中的所有文件节点
- Ansible 控制节点位于
`<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile`

当此文件存在时，该 ``beegfs_client`` 角色将自动检测并将其应用于客户端。



如果未使用 ``beegfs_client`` 角色配置客户端，则必须向每个客户端手动分发身份验证文件并在 ``beegfs-client.conf`` 文件中配置 ``connAuthFile`` 设置。从没有基于连接的身份验证的 BeeGFS 版本升级时，除非您在升级期间通过在 ``group_vars/ha_cluster.yml`` 中设置 ``beegfs_ha_conn_auth_enabled: false`` 来禁用基于连接的身份验证（不推荐），否则客户端将失去访问权限。

有关其他详细信息和备用配置选项，请参见 ["指定通用文件节点配置"](#) 节中的连接身份验证配置步骤。

升级到 BeeGFS v8

按照下面的步骤将您的 BeeGFS HA 集群从版本 7.4.6 升级到 BeeGFS v8。

概述

BeeGFS v8 引入了一些重大更改，这些更改需要在从 BeeGFS v7 升级之前进行其他设置。本文档指导您为 BeeGFS v8 的新要求准备集群，然后升级到 BeeGFS v8。



在升级到 BeeGFS v8 之前，请确保您的系统至少运行 BeeGFS 7.4.6。任何运行 BeeGFS 7.4.6 之前版本的群集必须首先["升级到版本 7.4.6"](#)，然后再继续此 BeeGFS v8 升级过程。

BeeGFS v8 的主要变化

BeeGFS v8 引入了以下主要更改：

- **许可证强制：** BeeGFS v8 需要许可证才能使用高级功能，如存储池、远程存储目标、BeeOND 等。在升级之前为您的 BeeGFS 集群获取有效的许可证。如果需要，您可以从 ["BeeGFS 许可证门户"](#) 获取临时 BeeGFS v8 评估许可证。
- **管理服务数据库迁移：** 要在 BeeGFS v8 中启用基于 TOML 的新格式配置，您必须手动将 BeeGFS v7 管理服务数据库迁移到更新的 BeeGFS v8 格式。
- **TLS 加密：** BeeGFS v8 引入 TLS 以实现服务之间的安全通信。您需要为 BeeGFS 管理服务和 `beegfs` 命令行实用程序生成和分发 TLS 证书，作为升级的一部分。

有关 BeeGFS 8 中的详细信息和其他更改，请参见 ["BeeGFS v8.0.0 升级指南"](#)。



升级到 BeeGFS v8 需要集群停机时间。此外，BeeGFS v7 客户端无法连接到 BeeGFS v8 集群。仔细协调集群和客户端之间的升级时间，以尽量减少对操作的影响。

为升级准备 BeeGFS 集群

在开始升级之前，请仔细准备您的环境，以确保平稳过渡并最大限度地减少停机时间。

1. 确保集群处于健康状态，所有 BeeGFS 服务都在其首选节点上运行。从运行 BeeGFS 服务的文件节点中，验证所有 Pacemaker 资源都在其首选节点上运行：

```
pcs status
```

2. 记录并备份集群配置。
 - a. 有关备份群集配置的说明，请参见 ["BeeGFS Backup 文档"](#)。
 - b. 备份现有管理数据目录：

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

- c. 从 beegfs 客户端运行以下命令并保存其输出以供参考：

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

- d. 如果使用镜像，请收集详细的状态信息：

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

3. 让您的客户为停机和停止 `beegfs-client` 服务做好准备。对于每个客户端，运行：

```
systemctl stop beegfs-client
```

4. 对于每个 Pacemaker 集群，禁用 STONITH。这将允许您在升级后验证集群的完整性，而不会触发不必要的节点重新启动。

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

5. 对于 BeeGFS 命名空间中的所有 Pacemaker 集群，使用 PCS 停止集群：

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

升级 BeeGFS 包

在集群中的所有文件节点上，为您的 Linux 发行版添加 BeeGFS v8 包存储库。有关使用官方 BeeGFS 存储库的说明，请参阅 ["BeeGFS 下载页面"](#)。否则，请相应地配置本地 beegfs 镜像存储库。

以下步骤演示如何在 RHEL 9 文件节点上使用官方 BeeGFS 8.2 存储库。在集群中的所有文件节点上执行以下步骤：

1. 导入 BeeGFS GPG 密钥：

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

2. 导入 BeeGFS 存储库：

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



删除任何先前配置的 BeeGFS 存储库，以避免与新的 BeeGFS v8 存储库发生冲突。

3. 清理您的包管理器缓存：

```
dnf clean all
```

4. 在所有文件节点上，将 BeeGFS 包更新为 BeeGFS 8.2。

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



在标准集群中，beegfs-mgmt 包将仅在前两个文件节点上更新。

升级管理数据库

在运行 BeeGFS 管理服务的其中一个文件节点上，执行以下步骤来将管理数据库从 BeeGFS v7 迁移到 v8。

1. 列出所有 NVMe 设备并筛选管理目标：

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

- a. 请注意输出中的设备路径。
- b. 将管理目标设备装载到现有管理目标装载点（将 /dev/nvmeXnY 替换为您的设备路径）：

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

2. 通过运行以下命令，将您的 BeeGFS 7 管理数据导入到新的数据库格式：

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

预期输出：

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



由于 BeeGFS v8 中更严格的验证要求，自动导入并非在所有情况下都能成功。例如，如果目标分配给不存在的存储池，则导入将失败。如果迁移失败，请勿继续升级。请联系 NetApp 支持人员以获得解决数据库迁移问题的帮助。作为临时解决方案，您可以降级 BeeGFS v8 包，并在解决问题的同时继续运行 BeeGFS v7。

3. 将生成的 SQLite 文件移动到管理服务装载：

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

4. 将生成的 `beegfs-mgmt.toml` 移动到管理服务装载：

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

准备 `beegfs-mgmt.toml` 配置文件将在完成下一节中的许可和 TLS 配置步骤后完成。

配置许可

1. 在运行 beegfs 管理服务的所有节点上安装 beegfs 许可证包。这通常是集群的前两个节点：

```
dnf install libbeegfs-license
```

2. 将您的 BeeGFS v8 许可证文件下载到管理节点并将其放在：

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

配置 TLS 加密

BeeGFS v8 需要 TLS 加密来实现管理服务和客户端之间的安全通信。在管理服务和客户端服务之间的网络通信上配置 TLS 加密有三个选项。推荐且最安全的方法是使用由受信任的证书颁发机构签名的证书。或者，您可以创建自己的本地 CA 来为 BeeGFS 群集签名证书。对于不需要加密或用于故障排除的环境，可以完全禁用 TLS，但不建议这样做，因为它会将敏感信息暴露给网络。

在继续之前，请按照[为 BeeGFS 8 配置 TLS 加密](#)指南中的说明为您的环境设置 TLS 加密。

更新管理服务配置

通过将设置从 BeeGFS v7 配置文件手动传输到 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.toml` 文件中来准备 BeeGFS v8 管理服务配置文件。

1. 在已装入管理目标的管理节点上，引用 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.conf` BeeGFS 7 的管理服务文件，然后将所有设置传输到 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.toml` 文件。对于基本设置，您的 `beegfs-mgmt.toml` 可能如下所示：

```

beemsg-port = 8008
grpc-port = 8010
log-level = "info"
node-offline-timeout = "900s"
quota-enable = false
auth-disable = false
auth-file = "/etc/beegfs/<mgmt_service_ip>_connAuthFile"
db-file = "/mnt/mgmt_tgt_mgmt01/data/mgmt.d.sqlite"
license-disable = false
license-cert-file = "/etc/beegfs/license.pem"
tls-disable = false
tls-cert-file = "/etc/beegfs/mgmt.d_tls_cert.pem"
tls-key-file = "/etc/beegfs/mgmt.d_tls_key.pem"
interfaces = ['i1b:mgmt_1', 'i2b:mgmt_2']

```

根据需要调整所有路径，以匹配您的环境和 TLS 配置。

2. 在运行管理服务的每个文件节点上，修改 systemd 服务文件以指向新的配置文件位置。

```

sudo sed -i 's|ExecStart=.*|ExecStart=nice -n -3
/opt/beegfs/sbin/beegfs-mgmt --config-file
/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.d.toml|'
/etc/systemd/system/beegfs-mgmt.service

```

- a. 重新加载 systemd:

```
systemctl daemon-reload
```

3. 对于运行管理服务的每个文件节点，为管理服务的 gRPC 通信打开端口 8010。

- a. 将端口 8010/tcp 添加到 beegfs 区域:

```
sudo firewall-cmd --zone=beegfs --permanent --add-port=8010/tcp
```

- b. 重新加载防火墙以应用更改:

```
sudo firewall-cmd --reload
```

更新 BeeGFS 监控脚本

Pacemaker beegfs-monitor OCF 脚本需要更新以支持新的 TOML 配置格式和 systemd 服务管理。更新集群中一个节点上的脚本，然后将更新后的脚本复制到所有其他节点。

1. 创建当前脚本的备份：

```
cp /usr/lib/ocf/resource.d/eseries/beegfs-monitor  
/usr/lib/ocf/resource.d/eseries/beegfs-monitor.bak.$(date +%F)
```

2. 将管理配置文件路径从 `.conf` 更新为 `.toml`：

```
sed -i 's|mgmt_config/beegfs-mgmt.d|.conf|mgmt_config/beegfs-mgmt.d.toml|'  
/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

或者，在脚本中手动找到以下块：

```
case $type in  
    management)  
        conf_path="${configuration_mount}/mgmt_config/beegfs-mgmt.d.conf"  
        ;;
```

并将其替换为：

```
case $type in  
    management)  
        conf_path="${configuration_mount}/mgmt_config/beegfs-mgmt.d.toml"  
        ;;
```

3. 更新 `get_interfaces()` 和 `get_subnet_ips()` 函数以支持 TOML 配置：

a. 在文本编辑器中打开脚本：

```
vi /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

b. 找到两个函数：`get_interfaces()` 和 `get_subnet_ips()`。

c. 删除两个完整函数，从 `get_interfaces()` 开始到 `get_subnet_ips()` 结束。

d. 复制并粘贴以下更新的函数来替换它们：

```

# Return network communication interface name(s) from the BeeGFS
resource's connInterfaceFile
get_interfaces() {
    # Determine BeeGFS service network IP interfaces.
    if [ "$type" = "management" ]; then
        interfaces_line=$(grep "^interfaces =" "$conf_path")
        interfaces_list=$(echo "$interfaces_line" | sed "s/.*= \[\\(.*/\)\]/\1/")
        interfaces=$(echo "$interfaces_list" | tr -d '"' | tr -d " " | tr
        ', ' '\n')

        for entry in $interfaces; do
            echo "$entry" | cut -d ':' -f 1
        done
    else
        connInterfacesFile_path=$(grep "^connInterfacesFile" "$conf_path"
        | tr -d "[:space:]" | cut -f 2 -d "=")

        if [ -f "$connInterfacesFile_path" ]; then
            while read -r entry; do
                echo "$entry" | cut -f 1 -d ':'
            done < "$connInterfacesFile_path"
        fi
    fi
}

# Return list containing all the BeeGFS resource's usable IP
addresses. *Note that these are filtered by the connNetFilterFile
entries.
get_subnet_ips() {
    # Determine all possible BeeGFS service network IP addresses.
    if [ "$type" != "management" ]; then
        connNetFilterFile_path=$(grep "^connNetFilterFile" "$conf_path" |
        tr -d "[:space:]" | cut -f 2 -d "=")

        filter_ips=""
        if [ -n "$connNetFilterFile_path" ] && [ -e
        $connNetFilterFile_path ]; then
            while read -r filter; do
                filter_ips="$filter_ips $(get_ipv4_subnet_addresses $filter)"
            done < $connNetFilterFile_path
        fi

        echo "$filter_ips"
    fi
}

```

- e. 保存并退出文本编辑器。
- f. 运行以下命令来检查脚本是否存在语法错误，然后再继续。无输出表示脚本语法正确。

```
bash -n /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

4. 将更新的 beegfs-monitor OCF 脚本复制到集群中的所有其他节点，以确保一致性：

```
scp /usr/lib/ocf/resource.d/eseries/beegfs-monitor  
user@node:/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

使集群恢复联机

1. 完成所有之前的升级步骤后，通过在所有节点上启动 BeeGFS 服务，使集群重新联机。

```
pcs cluster start --all
```

2. 验证 beegfs-mgmt 服务是否已成功启动：

```
journalctl -xeu beegfs-mgmt
```

预期输出包括以下行：

```
Started Cluster Controlled beegfs-mgmt.  
Loaded config file from "/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-  
mgmt.toml"  
Successfully initialized certificate verification library.  
Successfully loaded license certificate: TMP-113489268  
Opened database at "/mnt/mgmt_tgt_mgmt01/data/mgmt.sqlite"  
Listening for BeeGFS connections on [::]:8008  
Serving gRPC requests on [::]:8010
```



如果日志中出现错误，请检查管理配置文件路径，并确保所有值都已从 BeeGFS 7 配置文件正确传输。

3. 运行 `pcs status` 并验证集群是否正常运行，服务是否在其首选节点上启动。
4. 验证集群是否正常后，重新启用 STONITH：

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

5. 继续下一节以升级群集中的 BeeGFS 客户端，并检查 BeeGFS 群集的运行状况。

升级 BeeGFS 客户端

成功将群集升级到 BeeGFS v8 后，还必须升级所有 BeeGFS 客户端。

以下步骤概述了在基于 Ubuntu 的系统上升级 BeeGFS 客户端的过程。

1. 如果尚未完成，请停止 BeeGFS 客户端服务：

```
systemctl stop beegfs-client
```

2. 为您的 Linux 发行版添加 BeeGFS v8 包存储库。有关使用官方 BeeGFS 存储库的说明，请参阅["BeeGFS 下载页面"](#)。否则，请相应地配置本地 BeeGFS 镜像存储库。

以下步骤在基于 Ubuntu 的系统上使用官方 BeeGFS 8.2 存储库：

3. 导入 BeeGFS GPG 密钥：

```
wget https://www.beegfs.io/release/beegfs_8.2/gpg/GPG-KEY-beegfs -O  
/etc/apt/trusted.gpg.d/beegfs.asc
```

4. 下载存储库文件：

```
wget https://www.beegfs.io/release/beegfs_8.2/dists/beegfs-noble.list -O  
/etc/apt/sources.list.d/beegfs.list
```



删除任何先前配置的 BeeGFS 存储库，以避免与新的 BeeGFS v8 存储库发生冲突。

5. 升级 BeeGFS 客户端软件包：

```
apt-get update  
apt-get install --only-upgrade beegfs-client
```

6. 为客户端配置 TLS。使用 BeeGFS CLI 需要 TLS。请参阅["为 BeeGFS 8 配置 TLS 加密"](#)过程在客户端上配置 TLS。

7. 启动 BeeGFS 客户端服务：

```
systemctl start beegfs-client
```

验证升级

完成升级到 BeeGFS v8 后，运行以下命令以验证升级是否成功。

1. 验证根 inode 是否由与之前相同的元数据节点所有。如果您使用了管理服务中的 `import-from-v7` 功能，则应自动执行此操作：

```
beegfs entry info /mnt/beegfs
```

2. 验证所有节点和目标均处于联机状态且状态良好：

```
beegfs health check
```



如果"Available Capacity"检查警告目标可用空间不足，您可以调整 `beegfs-mgmt.d.toml` 文件中定义的"capacity pool"阈值，使其更适合您的环境。

升级HA集群中的PacMaker和Corosync软件包

按照以下步骤升级HA集群中的PacMaker和Corosync软件包。

概述

升级PacMaker和Corosync可确保集群受益于新功能、安全修补程序和性能改进。

升级方法

建议使用两种方法升级集群：滚动升级或完全关闭集群。每种方法都有自己的优缺点。升级过程可能因您的PacMaker版本而异。请参见ClusterLabs["升级PacMaker集群"](#)文档以确定使用哪种方法。在执行升级方法之前，请验证：

- NetApp BeeGFS解决方案支持新的PacMaker和Corosync软件包。
- 您的BeeGFS文件系统和PacMaker集群配置具有有效的备份。
- 集群处于运行状况良好的状态。

滚动升级

此方法包括从集群中删除每个节点、对其进行升级、然后将其重新引入集群、直到所有节点都运行新版本为止。这种方法可以使集群保持正常运行、这是大型HA集群的理想选择、但会带来在此过程中运行混合版本的风险。在双节点集群中应避免使用此方法。

1. 确认集群处于最佳状态、每个BeeGFS服务都在其首选节点上运行。有关详细信息、请参见 ["检查集群的状态"](#)。
2. 对于要升级的节点、请将其置于备用模式以耗尽(或移动)所有BeeGFS服务：

```
pcs node standby <HOSTNAME>
```

3. 运行以下命令、验证节点的服务是否已耗尽：

```
pcs status
```

确保未将备用节点上的服务报告为 Started。



根据您的集群大小、服务可能需要几秒钟或几分钟才能移至姊妹节点。如果BeeGFS服务无法在姊妹节点上启动，请参阅["故障排除指南"](#)。

4. 关闭节点上的集群：

```
pcs cluster stop <HOSTNAME>
```

5. 升级节点上的PacMaker、Corosync和pcs软件包：



软件包管理器命令因操作系统而异。以下命令适用于运行RHEL 8及更高版本的系统。

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

6. 在节点上启动PacMaker集群服务：

```
pcs cluster start <HOSTNAME>
```

7. 如果`pcs`软件包已更新、请向集群重新对节点进行身份验证：

```
pcs host auth <HOSTNAME>
```

8. 使用工具验证PacMaker配置是否仍然有效 `crm_verify`。



在集群升级期间、只需验证一次即可。

```
crm_verify -L -V
```

9. 使节点退出待机状态：

```
pcs node unstandby <HOSTNAME>
```

10. 将所有BeeGFS服务重新定位回其首选节点：

```
pcs resource relocate run
```

11. 对集群中的每个节点重复上述步骤、直到所有节点都运行所需的PacMaker、Corosync和pcs版本为止。

12. 最后、运行 `pcs status` 并验证集群是否运行正常、并 `Current DC` 报告所需的PacMaker版本。



如果 `Current DC` 报告"mxy-version"、则表示集群中的某个节点仍在运行先前的Pacemaker版本、需要进行升级。如果任何升级后的节点无法重新加入集群或资源无法启动、请查看集群日志、并参阅PacMaker发行说明或用户指南以了解已知的升级问题。

完全关闭集群

在这种方法中、所有集群节点和资源都会关闭、节点会升级、然后集群会重新启动。如果PacMaker和Corosync版本不支持混合版本配置、则必须使用此方法。

1. 确认集群处于最佳状态、每个BeeGFS服务都在其首选节点上运行。有关详细信息、请参见 ["检查集群的状态"](#)。
2. 关闭所有节点上的集群软件(PacMaker和Corosync)。



根据集群大小、整个集群可能需要几秒钟或几分钟才能停止。

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

3. 关闭所有节点上的集群服务后、根据您的要求升级每个节点上的PacMaker、Corosync和pcs软件包。



软件包管理器命令因操作系统而异。以下命令适用于运行RHEL 8及更高版本的系统。

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

4. 升级所有节点后、在所有节点上启动集群软件：

```
pcs cluster start --all
```

5. 如果 `pcs` 软件包已更新、请重新对集群中的每个节点进行身份验证：

```
pcs host auth <HOSTNAME>
```

6. 最后、运行 `pcs status` 并验证集群是否运行正常、并报告正确的 `Current DC` PacMaker 版本。



如果 `Current DC` 报告 "mxy-version"、则表示集群中的某个节点仍在运行先前的 Pacemaker 版本、需要进行升级。

更新文件节点适配器固件

按照以下步骤将文件节点的 ConnectX-7 适配器更新为最新固件。

概述

要支持新的 MLNX_OFED 驱动程序、启用新功能或修复错误、可能需要更新 ConnectX-7 适配器固件。本指南将使用 NVIDIA 的 `mlxfwmanager` 实用程序进行适配器更新、因为它易于使用且效率高。

升级注意事项

本指南介绍更新 ConnectX-7 适配器固件的两种方法：滚动更新和双节点集群更新。根据集群大小选择适当的更新方法。在执行固件更新之前、请验证：

- 安装了受支持的 MLNR_OFED 驱动程序，请参阅["技术要求"](#)。
- 您的 BeeGFS 文件系统和 PacMaker 集群配置具有有效的备份。
- 集群处于运行状况良好的状态。

固件更新准备

建议使用 NVIDIA 的 `mlxfwmanager` 实用程序更新节点的适配器固件、该固件与 NVIDIA 的 MLNR_OFED 驱动程序捆绑在一起。开始更新之前、请从下载适配器的固件映像["NVIDIA 的支持站点"](#)、并将其存储在每个文件节点上。



对于联想 ConnectX-7 适配器、请使用 `mlxfwmanager\_LES` NVIDIA 页面上提供的工具["OEM 固件"](#)。

滚动更新方法

对于具有两个以上节点的任何HA集群、建议使用此方法。此方法涉及一次更新一个文件节点上的适配器固件、以使HA集群能够继续处理请求、但建议避免在此期间处理I/O。

1. 确认集群处于最佳状态、每个BeeGFS服务都在其首选节点上运行。有关详细信息、请参见 ["检查集群的状态"](#)。
2. 选择要更新的文件节点并将其置于备用模式、此模式会从该节点中消耗(或移动)所有BeeGFS服务：

```
pcs node standby <HOSTNAME>
```

3. 运行以下命令、验证节点的服务是否已耗尽：

```
pcs status
```

确认没有服务在备用节点上报告为 Started。



根据集群大小、BeeGFS服务可能需要几秒钟或几分钟才能移至姊妹节点。如果BeeGFS服务无法在姊妹节点上启动，请参阅["故障排除指南"](#)。

4. 使用更新适配器固件 `mlxfwmanager`。

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

记下接收固件更新的 `PCI Device Name` 每个适配器的。

5. 使用实用程序重置每个适配器 `mlxfwreset` 以应用新固件。



某些固件更新可能需要重新启动才能应用更新。有关指导、请参见 ["NVIDIA的mxfwreset限制"](#) 如果需要重新启动、请执行重新启动、而不是重置适配器。

- a. 停止opensm服务：

```
systemctl stop opensm
```

- b. 对前面提到的每个执行以下命令 PCI Device Name。

```
mlxfwreset -d <pci_device_name> reset -y
```

- c. 启动opensm服务：

```
systemctl start opensm
```

- d. 重新启动 `eseries_nvme_ib.service`。

```
systemctl restart eseries_nvme_ib.service
```

- e. 验证 E 系列存储阵列的卷是否存在。

```
multipath -ll
```

1. 运行 `ibstat` 并验证所有适配器是否以所需的固件版本运行：

```
ibstat
```

2. 在节点上启动PacMaker集群服务：

```
pcs cluster start <HOSTNAME>
```

3. 使节点退出待机状态：

```
pcs node unstandby <HOSTNAME>
```

4. 将所有BeeGFS服务重新定位回其首选节点：

```
pcs resource relocate run
```

对集群中的每个文件节点重复上述步骤、直到所有适配器均已更新为止。

双节点集群更新方法

对于仅包含两个节点的HA集群、建议使用此方法。此方法类似于滚动更新、但还包括一些额外的步骤、以防止在一个节点的集群服务停止时出现服务停机。

1. 确认集群处于最佳状态、每个BeeGFS服务都在其首选节点上运行。有关详细信息、请参见 ["检查集群的状态"](#)。
2. 选择要更新的文件节点并将该节点置于备用模式、此操作将从该节点中消耗(或移动)所有BeeGFS服务：

```
pcs node standby <HOSTNAME>
```

3. 运行以下命令、验证节点的资源是否已耗尽：

```
pcs status
```

确认没有服务在备用节点上报告为 Started。



根据集群大小、BeeGFS服务可能需要数秒或数分钟才能报告为 `Started` 在姐妹节点上。如果BeeGFS服务无法启动，请参见["故障排除指南"](#)。

4. 将集群置于维护模式。

```
pcs property set maintenance-mode=true
```

5. 使用更新适配器固件 mlxfwmanager。

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

记下接收固件更新的 `PCI Device Name` 每个适配器的。

6. 使用实用程序重置每个适配器 `mlxfwreset` 以应用新固件。



某些固件更新可能需要重新启动才能应用更新。有关指导、请参见。["NVIDIA的mxfwreset限制"](#)如果需要重新启动、请执行重新启动、而不是重置适配器。

- a. 停止opensm服务：

```
systemctl stop opensm
```

- b. 对前面提到的每个执行以下命令 PCI Device Name。

```
mlxfwreset -d <pci_device_name> reset -y
```

- c. 启动opensm服务：

```
systemctl start opensm
```

7. 运行 `ibstat` 并验证所有适配器是否以所需的固件版本运行：

```
ibstat
```

8. 在节点上启动PacMaker集群服务：

```
pcs cluster start <HOSTNAME>
```

9. 使节点退出待机状态：

```
pcs node unstandby <HOSTNAME>
```

10. 使集群退出维护模式。

```
pcs property set maintenance-mode=false
```

11. 将所有BeeGFS服务重新定位回其首选节点：

```
pcs resource relocate run
```

对集群中的每个文件节点重复上述步骤、直到所有适配器均已更新为止。

升级E-Series存储阵列

请按照以下步骤升级HA集群的E系列存储阵列组件。

概述

确保HA集群的NetApp E-Series存储阵列始终使用最新固件、以确保最佳性能和更高的安全性。存储阵列的固件更新可通过SANtricity操作系统、NVSRAID和驱动器固件文件应用。



虽然可以在HA集群联机的情况下升级存储阵列、但建议在进行所有升级时将集群置于维护模式。

块节点升级步骤

以下步骤概述了如何使用 `Netapp\_Eseries.Santricity` Ands得以 收集来更新存储阵列的固件。在继续操作之前、请查看["升级注意事项"](#)以更新E-Series系统。



只能从11.70.5P1升级到SANtricity OS 11.80或更高版本。在应用进一步升级之前、必须先将存储阵列升级到11.70.5P1。

1. 验证您的SANtricity控制节点是否正在使用最新的Ants解决 方案集合。

- 用于升级可访问的集合 ["Ansible Galax河"](#)下，运行以下命令：

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

- 对于脱机升级，请从下载收集tarball"[Ansible Galaxy河](#)"，将其传输到控制节点，然后执行：

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

请参见 "[正在安装收集](#)" 有关详细信息 ...

2. 为存储阵列和驱动器获取最新固件。
 - a. 下载固件文件。
 - * SANtricity OS和NVSRAM:*导航到"[NetApp 支持站点](#)"并下载适用于您的存储阵列型号的SANtricity OS和NVSRAM:的最新版本。
 - *驱动器固件:*导航到"[E-Series磁盘固件站点](#)"并下载每个存储阵列驱动器型号的最新固件。
 - b. 将SANtricity操作系统、NVSRAM和驱动器固件文件存储在您的Ans负责人 控制节点的`<inventory\_directory>/packages`目录中。
3. 如有必要、请更新集群的"Andsess"清单文件、以包括所有需要更新的存储阵列(块节点)。有关指导、请参见"[Ansible清单概述](#)"一节。
4. 确保集群处于最佳状态、并且每个BeeGFS服务都位于其首选节点上。有关详细信息、请参见 "[检查集群的状态](#)"。
5. 按照中的说明将集群置于维护模式"[将集群置于维护模式](#)"。
6. 创建一个名为的新的Ands要用手册 update_block_node_playbook.yml。使用以下内容填充此操作手册、将SANtricity操作系统、NVSRAM并 将驱动器固件版本更换为所需的升级路径：

```
- hosts: eseries_storage_systems  
  gather_facts: false  
  any_errors_fatal: true  
  collections:  
    - netapp_eseries.santricity  
  vars:  
    eseries_firmware_firmware: "packages/<SantricityOS>.dlp"  
    eseries_firmware_nvram: "packages/<NVSRAM>.dlp"  
    eseries_drive_firmware_list:  
      - "packages/<drive_firmware>.dlp"  
    eseries_drive_firmware_upgrade_drives_online: true  
  
  tasks:  
    - name: Configure NetApp E-Series block nodes.  
      import_role:  
        name: nar_santricity_management
```

7. 要启动更新、请从您的Andsute控制节点执行以下命令：

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

8. 完成该操作手册后、验证每个存储阵列是否处于最佳状态。
9. 将集群移出维护模式、并验证集群是否处于最佳状态、并且每个BeeGFS服务都位于其首选节点上。

维护和维修

故障转移和故障恢复服务

在集群节点之间移动BeeGFS服务。

概述

BeeGFS服务可以在集群中的节点之间进行故障转移、以确保客户端能够在节点发生故障或您需要执行计划内维护时继续访问文件系统。本节介绍管理员在从故障中恢复后修复集群或在节点之间手动移动服务的各种方法。

步骤

故障转移和故障恢复

故障转移(计划内)

通常、当您需使单个文件节点脱机进行维护时、您需要从该节点移动(或消耗)所有BeeGFS服务。要实现此目的、可以先将节点置于备用状态：

```
pcs node standby <HOSTNAME>
```

验证后使用 `pcs status` 已在备用文件节点上重新启动所有资源、您可以根据需要关闭此节点或对其进行其他更改。

故障恢复(计划内故障转移之后)

准备好先将BeeGFS服务还原到首选节点时、请先运行 `pcs status` 并在"节点列表"中验证状态是否为备用。如果节点已重新启动、它将显示脱机、直到您使集群服务联机为止：

```
pcs cluster start <HOSTNAME>
```

节点联机后、使用以下命令将其置于备用状态：

```
pcs node unstandby <HOSTNAME>
```

最后、使用以下命令将所有BeeGFS服务重新定位回其首选节点：

```
pcs resource relocate run
```

故障恢复(计划外故障转移之后)

如果节点发生硬件或其他故障、HA集群应自动做出响应并将其服务移动到运行正常的节点、以便管理员有时间采取更正操作。在继续操作之前、请参考["故障排除"](#)一节以确定故障转移的原因并解决任何未解决的问题。在节点重新启动并运行正常后、您可以继续执行故障恢复。

如果节点在计划外(或计划内)重新启动后启动、则集群服务不会设置为自动启动、因此您首先需要使用以下命令使节点联机：

```
pcs cluster start <HOSTNAME>
```

接下来、清理任何资源故障并重置节点的隔离历史记录：

```
pcs resource cleanup node=<HOSTNAME>
pcs stonith history cleanup <HOSTNAME>
```

在中验证 `pcs status` 节点已联机且运行状况良好。默认情况下、BeeGFS服务不会自动进行故障恢复、以避免意外将资源移回运行状况不正常的节点。准备好后、使用以下命令将集群中的所有资源返回到其首选节点：

```
pcs resource relocate run
```

将单个**BeeGFS**服务移动到备用文件节点

将**BeeGFS**服务永久移动到新文件节点

如果要永久更改单个BeeGFS服务的首选文件节点、请调整Ansible清单、以便首先列出首选节点、然后重新运行Ansible攻略手册。

例如、在此示例文件中 `inventory.yml` 、`beegfs_01`是运行BeeGFS管理服务的首选文件节点：

```
mgmt:
  hosts:
    beegfs_01:
    beegfs_02:
```

反转顺序将使beegfs_02上的管理服务成为首选发生原因：

```
mgmt:
  hosts:
    beegfs_02:
    beegfs_01:
```

临时将**BeeGFS**服务移动到备用文件节点

通常、如果某个节点正在进行维护、您需要使用[故障转移和故障恢复步骤](#故障转移和故障恢复)将所有服务从该节点移出。

如果出于某种原因需要将单个服务移动到其他文件节点、请运行：

```
pcs resource move <SERVICE>-monitor <HOSTNAME>
```



请勿指定单个资源或资源组。请始终为要重新定位的BeeGFS服务指定监控器的名称。例如，要将BeeGFS管理服务移至beegfs_02，请运行： `pcs resource move mgmt-monitor beegfs_02`。可以重复执行此过程、将一个或多个服务从其首选节点移出。使用服务验证 `pcs status` 是否已在新节点上重新定位/启动。

要将BeeGFS服务移回其首选节点、请先清除临时资源限制(根据需要对多个服务重复此步骤)：

```
pcs resource clear <SERVICE>-monitor
```

然后、在准备好将服务实际移回其首选节点时、运行：

```
pcs resource relocate run
```

注意：此命令将重新定位不再具有临时资源限制的任何服务、而这些服务不位于其首选节点上。

将集群置于维护模式

防止HA集群意外对环境中的预期更改做出响应。

概述

将集群置于维护模式将禁用所有资源监控、并阻止Pacemaker移动或以其他方式管理集群中的资源。所有资源都将在其原始节点上保持运行状态、无论是否出现临时故障情况、从而使其无法访问。建议/有用的情形包括：

- 网络维护、可能会暂时中断文件节点与BeeGFS服务之间的连接。
- 块节点升级。
- 文件节点操作系统、内核或其他软件包更新。

通常、手动将集群置于维护模式的唯一原因是、防止集群对环境中的外部更改做出响应。如果集群中的单个节点

需要物理修复、请勿使用维护模式、只需按照上述操作步骤 将该节点置于备用状态即可。请注意、重新运行Ansible会自动将集群置于维护模式、以便于进行大多数软件维护、包括升级和配置更改。

步骤

要检查集群是否处于维护模式、请运行：

```
pcs property config
```

此 `maintenance-mode` 如果集群运行正常、则不会显示属性。如果集群当前处于维护模式，则该属性将报告为 `true`。要启用维护模式、请运行：

```
pcs property set maintenance-mode=true
```

您可以通过运行pcs状态并确保所有资源显示"(非受管)"来进行验证。要使集群退出维护模式、请运行：

```
pcs property set maintenance-mode=false
```

停止并启动集群

正常停止和启动HA集群。

概述

本节介绍如何正常关闭和重新启动BeeGFS集群。可能需要执行此操作的示例场景包括电气维护或在数据中心或机架之间迁移。

步骤

如果出于任何原因需要停止整个BeeGFS集群并关闭、则会运行所有服务：

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

也可以在单个节点上停止集群(这会 自动将服务故障转移到另一个节点)、但建议首先将该节点置于备用状态(请参见["故障转移"](#)一节)：

```
pcs cluster stop <HOSTNAME>
```

要在所有节点上启动集群服务和资源、请运行：

```
pcs cluster start --all
```

或使用以下命令在特定节点上启动服务：

```
pcs cluster start <HOSTNAME>
```

此时、请运行 `pcs status` 并验证集群和BeeGFS服务是否已在所有节点上启动、以及服务是否正在预期节点上运行。



根据集群大小、整个集群可能需要几秒钟或几分钟才能停止或显示为中已启动 `pcs status`。如果挂起超过五分钟、请在运行Ctrl+C取消命令之前登录到集群的每个节点、然后使用 ``pcs status`` 查看该节点上的集群服务(Corosync/PacMaker)是否仍在运行。``pcs cluster <COMMAND>`` 您可以从集群仍处于活动状态的任何节点上检查阻止集群的资源。手动输入问题描述 地址、然后命令应完成或可以重新运行以停止任何剩余服务。

替换文件节点

如果原始服务器出现故障、请更换文件节点。

概述

本节概述了更换集群中的文件节点所需的步骤。以下步骤假定文件节点因硬件问题描述 而失败、并替换为新的相同文件节点。

步骤

1. 物理更换文件节点、并还原到块节点和存储网络的所有布线。
2. 在文件节点上重新安装操作系统、包括添加Red Hat订阅。
3. 在文件节点上配置管理和BMC网络。
4. 如果主机名、IP、PCIe到逻辑接口映射或有关新文件节点的任何其他更改、请更新Ansible清单。通常、如果节点已更换为相同的服务器硬件、而您使用的是原始网络配置、则不需要执行此操作。
 - a. 例如、如果主机名发生更改、请创建(或重命名)节点的清单文件 (`host_vars/<NEW_NODE>.yaml``)、然后在Ansible清单文件中 (`inventory.yaml`)、将旧节点的名称替换为新节点名称：

```
all:
  ...
  children:
    ha_cluster:
      children:
        mgmt:
          hosts:
            node_h1_new:    # Replaced "node_h1" with "node_h1_new"
            node_h2:
```

5. 从集群中的其他一个节点中、删除旧节点：`pcs cluster node remove <HOSTNAME>`。



请勿在运行此步骤之前继续操作。

6. 在Ansible控制节点上：

a. 使用删除旧SSH密钥：

```
`ssh-keygen -R <HOSTNAME_OR_IP>`
```

b. 使用以下命令为替换节点配置无密码SSH：

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

7. 重新运行Ansible攻略手册以配置节点并将其添加到集群：

```
ansible-playbook -i <inventory>.yaml <playbook>.yaml
```

8. 此时、请运行 `pcs status` 并验证已更换的节点现在是否已列出并正在运行服务。

扩展或缩减集群

在集群中添加或删除组件。

概述

本节介绍了调整BeeGFS HA集群大小的各种注意事项和选项。通常、可以通过添加或删除组件来调整集群大小、这些组件通常是将两个文件节点设置为HA对的组件。如果需要、还可以添加或删除单个文件节点(或其他类型的集群节点)。

向集群添加组件

注意事项

通过添加更多组件来扩展集群是一个简单的过程。开始之前、请牢记每个HA集群中集群节点的最小和最大数量限制、并确定是否应向现有HA集群添加节点或创建新的HA集群。通常、每个组件都包含两个文件节点、但每个集群的最小节点数为三个节点(用于建立仲裁)、建议的最大值为十个(已测试)。对于高级情形、可以在部署双节点集群时添加一个不运行任何BeeGFS服务的“Tiebreaker”节点。如果您考虑进行此类部署、请联系NetApp支持部门。

在决定如何扩展集群时、请记住这些限制以及任何预期的未来集群增长。例如、如果您有一个六节点集群、并且需要再添加四个节点、则建议您只启动一个新的HA集群。



请记住、一个BeeGFS文件系统可以由多个独立的HA集群组成。这样、文件系统就可以继续扩展、远远超出底层HA集群组件的建议/硬限制。

步骤

向集群添加组件时、您需要 `host\_vars` 为每个新文件节点和块节点(E-Series阵列)创建文件。需要将这些主机的名称以及要创建的新资源添加到清单中。`group\_vars` 需要为每个新资源创建相应的文件。["使用自定义架构"](#)有关详细信息、请参见一节。

创建正确的文件后、只需使用命令重新运行自动化即可：

```
ansible-playbook -i <inventory>.yaml <playbook>.yaml
```

从集群中删除组件

在需要停用组件时、需要牢记一些注意事项、例如：

- 此组件正在运行哪些BeeGFS服务？
- 是否只是文件节点停用、块节点应附加到新的文件节点？
- 如果停用整个组件、数据应移至新的组件、分散到集群中的现有节点中、还是移动到新的BeeGFS文件系统或其他存储系统？
- 是否可以在中断期间发生这种情况、或者是否应在无中断的情况下进行？
- 该组件是否正在使用中、或者它是否主要包含不再处于活动状态的数据？

由于可能的起始点和所需的最终状态各不相同、因此请联系NetApp支持部门、以便我们能够根据您的环境和要求确定并帮助实施最佳策略。

故障排除

对BeeGFS HA集群进行故障排除。

概述

本节将介绍如何调查各种故障以及在运行BeeGFS HA集群时可能出现的其他情形并对其进行故障排除。

故障排除指南

正在调查意外故障转移

当节点意外隔离并将其服务移至另一个节点时、第一步应是查看集群是否在底部指示任何资源故障 `pcs status`。如果成功完成隔离且资源已在另一节点上重新启动、则通常不会显示任何内容。

通常、下一步是使用搜索整个systemd日志 `journalctl` 在其余任何一个文件节点上(所有节点上的Pacemaker日志均会同步)。如果您知道发生故障的时间、则可以在发生故障之前开始搜索(通常建议至少提前十分钟)：

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>"
```

以下各节显示了您可以在日志中查看的常见文本、以进一步缩小调查范围。

第1步：检查BeeGFS监控器是否检测到故障：

如果故障转移是由BeeGFS监控器触发的、则应看到错误(如果不是、请继续执行下一步)。

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i unexpected
[...]
```

```
Jul 01 15:51:03 beegfs_01 pacemaker-schedulerd[9246]: warning: Unexpected
result (error: BeeGFS service is not active!) was recorded for monitor of
meta_08-monitor on beegfs_02 at Jul 1 15:51:03 2022
```

在本示例中、BeeGFS服务meta_08因某种原因停止。要继续进行故障排除，我们应启动beegfs_02并查看位于的服务日志 /var/log/beegfs-meta-meta_08_tgt_0801.log。例如、BeeGFS服务可能由于内部问题描述 或节点问题而遇到应用程序错误。



与Pacemaker中的日志不同、BeeGFS服务的日志不会分发到集群中的所有节点。要调查这些类型的故障、需要原始节点中发生故障的日志。

监控器可能报告的问题包括：

- 无法访问目标！
 - 问题描述：表示无法访问块卷。
 - 故障排除：
 - 如果此服务也无法在备用文件节点上启动、请确认块节点运行状况良好。
 - 检查是否存在任何可能阻止从此文件节点访问块节点的物理问题、例如、InfiniBand适配器或缆线出现故障。
- 无法访问网络！
 - 问题描述：客户端用于连接到此BeeGFS服务的适配器均未联机。
 - 故障排除：
 - 如果多个/所有文件节点受到影响、请检查用于连接BeeGFS客户端和文件系统的网络是否存在故障。
 - 检查是否存在任何可能阻止从此文件节点访问客户端的物理问题、例如InfiniBand适配器或缆线出现故障。
- BeeGFS服务未处于活动状态！
 - 问题描述：BeeGFS服务意外停止。
 - 故障排除：
 - 在报告错误的文件节点上、检查受影响的BeeGFS服务的日志以查看它是否报告崩溃。如果发生这种情况、请向NetApp支持部门创建一个案例、以便可以调查崩溃情况。
 - 如果BeeGFS日志中未报告任何错误、请检查日志以查看systemd是否记录了服务停止的原因。在某些情况下、BeeGFS服务可能没有机会在进程终止之前记录任何消息(例如、有人运行时) kill -9 <PID>)。

第2步：检查节点是否意外离开集群

如果节点发生了一些灾难性硬件故障(例如、系统主板已损坏)或发生内核崩溃或类似的软件问题描述、BeeGFS监控器将不会报告错误。而是查找主机名、此时您应看到Pacemaker发出的消息、指出节点意外丢失：

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i <HOSTNAME>
[...]
```

```
Jul 01 16:18:01 beegfs_01 pacemaker-attrd[9245]: notice: Node beegfs_02
state is now lost
Jul 01 16:18:01 beegfs_01 pacemaker-controld[9247]: warning:
Stonith/shutdown of node beegfs_02 was not expected
```

第3步：验证Pacemaker是否能够隔离节点

在所有情况下、您都应看到Pacemaker尝试隔离节点、以验证其是否实际脱机(确切消息可能因隔离的发生原因而异)：

```
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]: warning: Cluster
node beegfs_02 will be fenced: peer is no longer part of the cluster
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]: warning: Node
beegfs_02 is unclean
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]: warning: Scheduling
Node beegfs_02 for STONITH
```

如果隔离操作成功完成、您将看到如下消息：

```
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]: notice: Operation 'off'
[2214070] (call 27 from pacemaker-controld.9247) for host 'beegfs_02' with
device 'fence_redfish_2' returned: 0 (OK)
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]: notice: Operation 'off'
targeting beegfs_02 on beegfs_01 for pacemaker-
controld.9247@beegfs_01.786df3a1: OK
Jul 01 16:18:14 beegfs_01 pacemaker-controld[9247]: notice: Peer
beegfs_02 was terminated (off) by beegfs_01 on behalf of pacemaker-
controld.9247: OK
```

如果隔离操作因某种原因失败、则BeeGFS服务将无法在另一个节点上重新启动、以避免出现数据损坏的风险。如果隔离设备(PDU或BMC)不可访问或配置不当、则该问题描述 将单独进行调查。

解决失败的资源操作(位于**pcs**状态的底部)

如果运行BeeGFS服务所需的资源发生故障、BeeGFS监控器将触发故障转移。如果发生这种情况，底部可能不会列出“失败的资源操作” pcs status，您应参阅有关如何操作的步骤[“计划外故障转移后的故障恢复”](#)。

否则、通常只能在两种情况下看到“资源操作失败”。

场景1: 检测到使用隔离代理的临时或永久问题描述、并且已重新启动或将其移动到其他节点。

某些隔离代理比其他隔离代理更可靠、每个隔离代理都将实施自己的监控方法、以确保隔离设备已准备就绪。特别是、我们发现Redfish隔离代理报告的资源操作失败、即使它仍显示为已启动:

```
* fence_redfish_2_monitor_60000 on beegfs_01 'not running' (7):
call=2248, status='complete', exitreason='', last-rc-change='2022-07-26
08:12:59 -05:00', queued=0ms, exec=0ms
```

如果隔离代理报告某个特定节点上的资源操作失败、则不会触发该节点上运行的BeeGFS服务的故障转移。只需在同一节点或不同节点上自动重新启动即可。

解决步骤:

1. 如果隔离代理始终拒绝在全部或部分节点上运行、请检查这些节点是否能够连接到隔离代理、并验证是否已在Ansible清单中正确配置隔离代理。
 - a. 例如、如果Redfish (BMC)隔离代理与它负责隔离的节点运行在同一个节点上、而操作系统管理和BMC IP位于同一个物理接口上、则某些网络交换机配置将不允许这两个接口之间进行通信(以防止网络环路)。默认情况下、HA集群会尝试避免在其负责隔离的节点上放置隔离代理、但在某些情形/配置中可能会发生这种情况。
2. 解决所有问题后(或者如果问题描述 似乎是临时的)、请运行 `pcs resource cleanup` 重置失败的资源操作。

场景2: BeeGFS监控器检测到问题描述 并触发故障转移、但出于某种原因、资源无法在二级节点上启动。

如果已启用隔离且未阻止资源在原始节点上停止(请参见"备用(故障)"故障排除部分)、则最可能的原因包括在二级节点上启动资源时出现问题、因为:

- 二级节点已脱机。
- 物理或逻辑配置问题描述 会阻止二级系统访问用作BeeGFS目标的块卷。

解决步骤:

1. 对于失败的资源操作中的每个条目:
 - a. 确认失败的资源操作为启动操作。
 - b. 根据出现故障的资源操作中指示的资源 and 指定的节点:
 - i. 查找并更正可能会阻止节点启动指定资源的任何外部问题。例如、如果BeeGFS IP地址(浮动IP)无法启动、请确认至少有一个所需接口已连接/联机并已连接到正确的网络交换机。如果BeeGFS目标(块设备/E系列卷)发生故障、请验证与后端块节点的物理连接是否按预期连接、并验证块节点是否运行正常。
 - c. 如果没有明显的外部问题、并且您希望在此事件中使用根发生原因、建议您先与NetApp支持部门一起创建案例进行调查、然后再继续操作、因为以下步骤可能会使根发生原因 分析(Root Analysis、RCA)变得极具挑战性/不可能。
2. 解决任何外部问题后:

- a. 从Ansible inventory.yml文件中注释掉所有无法正常工作的节点、然后重新运行完整的Ansible攻略手册、以确保在二级节点上正确设置所有逻辑配置。
 - i. 注意：在节点运行状况良好且您准备好进行故障恢复后、请勿忘记取消对这些节点的注释并重新运行攻略手册。
- b. 或者、您也可以尝试手动恢复集群：
 - i. 使用以下命令将所有脱机节点重新联机：`pcs cluster start <HOSTNAME>`
 - ii. 使用以下命令清除所有失败的资源操作：`pcs resource cleanup`
 - iii. 运行pcs状态并验证所有服务是否按预期启动。
 - iv. 如果需要、请运行 `pcs resource relocate run` 将资源移回其首选节点(如果有)。

常见问题

BeeGFS服务在请求时不会进行故障转移或故障恢复

可能的问题描述：`pcs resource relocate` 运行命令已执行、但从未成功完成。

*如何检查：*运行 `pcs constraint --full` 并检查ID为的任何位置约束 `pcs-relocate-<RESOURCE>`。

*如何解决：*运行 `pcs resource relocate clear` 然后重新运行 `pcs constraint --full` 以验证是否已删除额外的约束。

禁用隔离后、处于pcs状态的一个节点将显示"**standby (on-fail)**"

可能的问题描述：Pacemaker无法成功确认故障节点上的所有资源均已停止。

如何解决：

1. 运行 `pcs status` 并检查输出底部是否存在未"启动"或显示错误的任何资源、并解决任何问题。
2. 要使节点恢复联机、请运行 `pcs resource cleanup --node=<HOSTNAME>`。

在发生意外故障转移后、如果启用了隔离、则资源将以pcs状态显示"**started (on-fail)**"

*可能的问题描述：*发生了一个问题、触发了故障转移、但Pacemaker无法验证节点是否已隔离。发生这种情况的原因可能是隔离配置不当或存在具有隔离代理的问题描述 (例如：PDU已与网络断开连接)。

如何解决：

1. 验证节点是否已实际关闭。



如果您指定的节点实际上未关闭、但正在运行集群服务或资源、则会发生数据损坏/集群故障。

2. 手动确认隔离：`pcs stonith confirm <NODE>`

此时、服务应完成故障转移并在另一个运行正常的节点上重新启动。

常见故障排除任务

重新启动单个BeeGFS服务

通常、如果需要重新启动BeeGFS服务(例如、为了便于更改配置)、应通过更新Ansible清单并重新运行攻略手册来完成此操作。在某些情况下、可能需要重新启动单个服务以加快故障排除速度、例如更改日志记录级别而无需等待整个攻略手册运行。



除非同时将任何手动更改添加到Ansible清单中、否则将在下次运行Ansible攻略手册时还原这些更改。

选项1：由系统d控制的重新启动

如果存在BeeGFS服务无法使用新配置正确重新启动的风险、请先将集群置于维护模式、以防止BeeGFS监控器检测到服务已停止并触发不需要的故障转移：

```
pcs property set maintenance-mode=true
```

如果需要、可通过对服务配置进行任何更改 `/mnt/<SERVICE_ID>/_config/beegfs-.conf` (例如：`/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf`)、然后使用systemd重新启动它：

```
systemctl restart beegfs-*@<SERVICE_ID>.service
```

示例 `systemctl restart beegfs-meta@meta_01_tgt_0101.service`

选项2：起搏器控制的重新启动

如果您不关心新配置是否可能发生原因 会使服务意外停止(例如、更改日志记录级别)、或者您处于维护窗口而不担心停机、则只需为要重新启动的服务重新启动BeeGFS监控器即可：

```
pcs resource restart <SERVICE>-monitor
```

例如、要重新启动BeeGFS管理服务、请执行以下操作：`pcs resource restart mgmt-monitor`

版权信息

版权所有 © 2026 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。